NA 132 + 64 STRANE

PERSONALNI

računari

100

MAJ 1994.

CENA 5 NDIN

100 JUBILARNI BROJ

NOVA NAGRADNA IGRA

50.000 DIN

U HARDVERU I SOFTVERU

SUPERPREMIJA

JITEX

PENTIUM PC

PRVA NAGRADA

OTC COMPUTER

NOTEBOOK 486DX

NAŠI TESTOVI

DECpc ALPHA

TVRDI DISKOVI

EXCEL 5.0

WORD PERFECT FOR WINDOWS 6.0

BORLAND C++ 4.0

SPECIJALNI DODATAK NA 64 STRANE

WINDOWS 3.1

KOMPLETNO UPUTSTVO ZA UPOTREBU





100
JUBILARNI
BROJ

PERSONALNI **računari 100**

SADRŽAJ

SADRŽAJ OGLASA

časopis za informatiku
i računarstvo
YU ISSN 0352-7221
Izdaje BIGZ
maj 1994 Cena: 5 NDIN

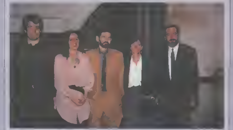
Izdaje i štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar Vojvode Mišića 17
Generalni direktor
Ilija Rapaić
Glavni i odgovorni urednik
Jovan Regasek
Zamenik glavnog i odgovornog urednika
Ranka Jovanović
Komercijalni urednik
Vesna Jeremić
Tehnički urednik
Radoje Kavedžić
Stručna redakcija
Nenad Batočanin (baze podataka), Dejan Ristanović (softver), Vladimir Stamenović (radne tabele), Vesna Ćosić (aktuelnosti), Zoran Životić (softver)
Adresa redakcije
11000 Beograd
Bulevar vojvode Mišića 17/III
Telefoni
647 476 (glavni urednik)
653-748 (redakcija)
651-666 (centrala, pretplata 226, stari brojevi 259)
Agencija BIGZ-a
(011) 651 793, 653-565
Telefaks
(011) 648-140, 647-565
Opšti poslovi plasmana
(011) 652 403, 651-666 / 277
Poslovi tiraža i kontrole
(011) 650-834, 647 533
Pretplata
(011) 650-528, 651 666 / 226, 259
Pretplata za zemlju
Za 6 meseci (šest brojeva) **24 NDIN**
Za 12 meseci (deset brojeva) **48 NDIN**
(na žiro-račun: D.P. BIGZ, 40806-603-6-23264)
Pretplata za inostranstvo
Za jednu godinu 80 DEM
(na devizni račun: D.P. BIGZ 40811-620-16101 820761-989-03377)
Zbog usporenog prenošenja uplata preko banke molimo pretplatnike da nam posle svake nove uplate odmah pošalju foto-kopiju uplatnice.
Rukopisi se ne vraćaju.

SEZAM **(011) 648-422 (15 linija)**
sistem za modemske komunikacije
Upravnik sistema
Jovan Regasek
Administratori na sistemu
Zoran Životić, Dejan Ristanović

HVALA, NIKAD BOLJE

Nismo imali sreću da se dva naša značajna jubileja – deset godina od osnivanja lista i stoti broj časopisa „Računari" poklope u isti dan, i stoga sam doveden u pomalo neprijatnu situaciju da u razmaku od samo nekoliko meseci pišem novi uvodnik na potpuno istu temu. Umesto da ponavljam stvari koje su tamo rečene, osećam potrebu da nastavim upravo tamo gde sam bio stao.

Taj decembar u kome smo pripremali jubilarni januarski broj koji je izlazio na desetogodišnjicu lista, bio je možda najtragičniji mesec u istoriji ovog časopisa. Tog meseca je naš izdavač Beogradski izdavačko-grafički zavod doneo odluku da povuče „Računare" i sa zadnjeg kioska u zemlji. Bio je to poslednji završni čin u višemesečnoj drami povlačenja lista sa novinskih kioska u sopstvene knjižare. Svaki izdavač naravno odlično zna da je novinama mesto na kiosku a knjigama u knjižarama, zna to i BIGZ ali mi naprosto nismo imali izbora. U uslovima koji su tada vladali na tržištu svaki primerak koji je upućen na kioske je bio zapravo poklonjen prodajnoj mreži.



Najuže jezgro časopisa „Računari": Dejan Ristanović, autor koji je objavio u listu gotovo 550 tekstova, Vesna Jeremić, komercijalni urednik, koja je organizovala dve najveće nagradne igre u istoriji lista, Jovan Regasek, glavni i odgovorni urednik, Ranka Jovanović, zamenik glavnog urednika i Zoran Životić, autor Sezama i najpopularniji programer u Jugoslaviji. Fotografija je snimljena novembra 1993. godine na proslavi Sezamovog rođendana u hotelu „Hyatt". Žao nam je što na njoj nisu i naši autori. Jer da nije njih, ne bi bilo ni „Računara".

Izrazio sam u tom uvodniku uverenje da će čitaoci uvek naći načina da dođu do lista – ako im je on potreban. Dok je čitalaca koji sa uzbuđenjem očekuju svaki novi broj pisalo je na kraju biće i „Računara". U tom trenutku međutim nismo ni slutili da i Redakciju i čitaoce očekuju mnogo gora iskušenja. Prvih dana januara „Računari" su mogli da se kupe jedino u BIGZ-ovoj prodavnici u Bulevaru vojvode Mišića i jedino za takozvane „poene". Nekome se to nije dopalo, okrenuo je telefon tržišne inspekcije i inspektorka (koja se predstavila kao kupac) je zabranila prodaju lista. Čitaoci su međutim i u takvim nemogućim okolnostima pronalazili načina da dođu do „Računara".

Dogodilo se na ponovio se! Prilike na tržištu su se malo sredile i mi svoj jubilarni stoti broj dočekujemo kao u najboljim danima. Sa bitno povećanim tiražem, sa obiljem tekstova i oglasa na 132 strane, sa umetkom *Windows 3.1* na 64 strane i, najzad, sa šansom da u velikoj već petoj po redu nagradnoj igri, većoj od svih koje smo imali do sada, osvojite PC računar Pentium klase 486DX

Redakcija Računara zahvaljuje Aleksandru Radovanoviću Pretorija (Južna Afrika) na tehničkom doprinosu u realizaciji ovog broja. Kompletna komunikacija na relaciji Beograd/Johannesburg išla je preko Internet noda Opsrey u Pretoriji, na kome je Aca sistem administrator, i FON-a u Beogradu, kome se takođe najlepše zahvaljujemo na pomoći.

notebook ili neku od osamdesetak takođe vrednih nagrada.

U ovom broju smo takođe uspeli da okupimo i nekoliko autora iz stare garde, uglavnom sa raznih strana sveta, čiji su tekstovi svojevremeno rado čitani u Računarima a nisu više aktivni u našem časopisu. Jelena Rupnik iz svog ugla baca još jedan kosi pogled na YU kompjutersku scenu, a Neverka Spalević nas podseća da je u našem obrazovanju još uvek sve po starom. Duško Savić, jedini mozak programiranja u našoj ekipi, pripremio je temu upravo iz svog faha, dok Jovan Skuljan ostaje veran asembleru za Intelove procesore. Vlada Kostić, *enfant terrible* časopisa „Računari" pokazuje da se ni u Americi nije puno promenio, dok je Aleksandar Radovanović počeo profesionalno da se bavi računarskim mrežama. Vratio se Miroslav Olenjin, a Josip Modk se prvi put javlja sa svojim tekstom.

Između onih koji su bili na početku i onih koji dolaze sada stoji, naravno, naša stalna autorska ekipa koja je i ovoga puta ponela glavni teret (u abecednom redosledu): Vladan Aleksić, Jovan Bulajić, Zoran D. Brkić, Viktor Cerovski, Vesna Ćosić, Dejan Jelović, Mijan Jovanović, Zoran Kehler, Aleksandar Kragl, Nenad Batočanin, Pavle Peković, Bojan Petrović, Jovan Regasek, Dejan Ristanović, Vladimir Stamenović, Dejan Veselinović, Zoran Životić. Nikada u istoriji, čini mi se, nismo imali toliko jakih autorskih imena na jednom mestu.

Sve je dakle ponovo kao nekad. Jedino je tehnologija otišla napred, ali to je već druga priča. Iz te druge priče ipak izdvajamo jedan detalj koji će vam možda biti zanimljiv. Ovaj broj Računara je pripremljen na krajnje neobičan način: Deo redakcije (Ranka Jovanović, Vesna Jeremić i Radoje Kavedžić) je sedeo u Beogradu, deo (glavni urednik) hiljadama kilometara odatle u Johannesburgu, dok su saradnici praktično bili rasuti na svim stranama sveta. Rukopisi su svakodnevno razmenjivani preko Interneta, a u kritičnim trenucima, kad nije bilo dovoljno vremena za pomalo spore Internet kanale, i direktnom modemskom vezom. Ovo je verovatno jedinstven slučaj u našem novinarstvu da broj pripremi tako razuđena redakcija čiji su članovi i saradnici na različitim kontinentima, i da uz to još izađe gotovo na vreme. Broj koji držite u rukama je plod novog vremena, u kome čitav svet povezan računarskim mrežama i „informacionim autoputevima" zaista postaje jedno – selo.

Jovan Regasek

Na osnovu Mišljenja Sekretarijata za informacije SR Srbije broj 413-01 10 od 4. februara 1991. godine plaća se osnovni porez na promet po povlašćenoj tarifi od 7 odsto


ADACOM
VAŠ SIGURAN PARTNER
STATVS
...IZVOĐENJE...ODRŽAVANJE...REVERZIBILNI
...ODRŽAVANJE...REVERZIBILNI INŽENJERING...KONSALT
INŽENJERING TEL. 34 14 96, TEL./FAX 3373 67
RAČUNARI. PRODAJA TEL. 62 92 33

SOFTVER

Tekst-procesori

WordPerfect za spore računare

Kada su predstavljeni *WordPerfect 6.0* za DOS i *Windows* zvaničnici kompanije su najavili i verziju koja će više odgovarati korisnicima čiji kompjuteri nemaju dovoljno snage za tako veliki softver. Obećanje je ispunjeno i pred nama je *WordPerfect 5.1+*.

Kao što se dalo očekivati, radi se o svojevrsnom hibridu između verzija 5.1 i 6.0, naravno za DOS okruženje. Najvažnija osobina novog *WordPerfect*-a je da ravnopravno može raditi sa svim formatima dokumenata koji su vremenom menjani sa verzijama ovog tekst-procesora. To znači da je podržan i format iz verzije 6.0. Osim toga, i *WordPerfect*-a 5.1+ kao i 6.0 omogućava da se iz samog tekst-procesora upravlja faks/modem uređajem.

WordPerfect 5.1+ dobio je bolji fajl menadžer koji omogućava sortiranje datoteka po datumu, vremenu kreiranja i veličini. Već od ranije velikom broju opcija koje pruža ovaj tekst procesor dodata je i elektronska pošta, a da bi je koristili na računaru more biti instaliran i paket *WordPerfect Office*. Iz verzije 6.0 preuzeti su i novi drajveri za mnoštvo raznih uređaja poput štampača, skenera ili faksova. Ako često razmenjujete dokumente sa korisnicima drugih tekst procesora, obradovaće vas alatka za konverziju tekstova koja prepoznaje sledeće formate: *WordPerfect 3.0 for Macintosh, Microsoft Word 2.0 for Windows, Lotus Ami Pro 3.0 for Windows, Microsoft Word 5.5 for DOS* i poslednje verzije programa *Multimate, Displaywrite, Professional Write, Wordstar, Microsoft Windows Write, Officewriter* i *Xywrite*.

U paketu sa tekst procesorom nalaze se i dva korisna programa. Prvi je *Screenextender Special Edition* koji vam omogućava da na ekranu lakše operišete sa proporcionalnim, skalabilnim i sabijenim slovnim likovima. Takođe, ovaj svojevrsni WYSIWYG dodatak pravilno tretira i atribute teksta kao što su bold, italik ili podvučena slova. Drugi program je *Bitstream Face Lift* sa jedanaest vektorskih slovnih likova. Naše korisnike obradovaće vest da ovaj program omogućava kreiranje novih slovnih likova, što daje nadu da postoji rešenje za naša slova i u samom paketu.

WordPerfect 5.1+ ima iste hardverske zahteve kao i prethodne verzije, odnosno može da radi na svemu što se naziva PC – od XT mašine do onih sa moćnim *Pentium*-om. Na disku zauzima oko 5 megabajta, a minimum operativne memorije koji je potreban za rad je 640 kilobajta.

Kontakt adresa: WordPerfect Corp. tel. 800/451-5151

Tržište

Kad se vrbi rodi grožđe

Hermes, Enterprise Messaging Server, Chicago – promocija ovih paketa je toliko puta odlagana da se već odavno pričaju vicevi o tome kako iz Microsoftovih pesoftverskih garnitura (*suites*) umesto nepostojećeg softvera ispadaju „ceduljice" sa pravdanjem za kašnjenja i novim obećanjima – u koje su i najnaivniji odavno prestali da veruju.

Uz Microsoft na listi „nepopravljivih" vode Novell i Apple, a pri samom vrhu su i Hewlett Packard, Motorola, WordPerfect, Gaia, SCO i AT&T. Svi su oni odavno položili zgovor po kojem „kasne jer ne mogu sebi dopustiti da puste na tržište nedorađene proizvode" – sve je to lepo, ali bi lepo bilo i da se potrude da malo realnije procene sopstvene mogućnosti. *Hermes* kasni već gotovo godinu dana, EMS gotovo dve. Izlazak *Chicaga* je sa početka pomeren na drugu polovinu ove godine. *Daytona* (ili NT 3.5) koji je najavljen za 30. jun, ali je već očigledno da će rok biti probijen.

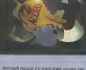
Zanimljivo je da prodavci softvera na ovo gledaju prilično blagonaklone i malo je reći da su u prihvatanju novoga prilično konzervativni – niko želi da rizikuje prodajući nedorađene i neproverene verzije, pogotovo ne one sa oznakom 1.0. Naravno, ima i onih koji zbog toga trpe – jedan veliki prodavac u Britaniji se vajka da zbog kašnjenja *Hermesa* nije mogao valjano da promoviše *Windows NT* (kažu da je *Hermes* u stvari bio spreman za tržište, ali ga je Microsoft zaustavio jer se bolje izvršava u operativnom sistemu *Daytona*). Isti prodavac dodaje da ova anju DOS operativnoći da čeka umesto *Hermesa* spas za NT će potražiti u Lotusovom programu *Notes*.

Baze podataka

Microsoft Access: Preko 300 novih opcija

Microsoft Access je već sa prvom verzijom dobio epitet najboljeg paketa za upravljanje bazama podataka koji radi u *Windows* radnom okruženju. Kao što je to običaj, verzija 2.0 koja je pred nama donosi niz poboljšanja i novih opcija. Pedantni statističari u Microsoft-u su ih prebrojali i spisak ima preko tri stotine stavki.

Za očekivati je da se najveći broj novosti odnosi na povećanja produktivnosti kod kreiranja baza i pripreme izveštaja. Naravno, sve je izvedeno kroz već popularne „čarobnjake" (*Wizards*). *Access 2.0* ima preko trideset novih *Wizards* alatki, uključujući revolucionarni *Table Wizard*. Sada je dovoljno odgovoriti na samo nekoliko pitanja, kliknuti na *Finish* dugme i nova tabela je pred vama.



Microsoft Access 2.0 predstavlja prirodni deo *Microsoft Office* paketa. Prenos podataka iz ovog programa u, na primer, *Word 6.0* je potpuno automatizovan i odvija se posredstvom OLE 2.0 tehnike.

Prethodne verzije ovog paketa prodate su u preko milion kopija zahvaljujući pre svega činjenici da su se velike korporacije odlučile da baš ona bude standardni alat za održavanje baza podataka. Zbog toga je Microsoft istovremeno pustio na tržište *Access Developer Toolkit*, paket razvojnih alatki namenjen korporacijskim projektantima softvera. Pomoću njega moguće je kreirati specifične aplikacije koje se u svom radu oslanjaju na *Access 2.0*.

U paketu *Access Developer Toolkit* nalaze se *run time* verzija *Access 2.0* neophodna za pokretanje samostalnih aplikacija, *Setup Wizard*, program koji na lak način omogućava kreiranje instalacionih disketa, *Microsoft Access 2.0 Database Engine*, skup rutina za povezivanje sa *Visual Basic*-om, kompletne informacije na osnovu kojih možete praviti sopstvene „čarobnjake". Mesta se našlo i za bogatu biblioteku primera u izvornom kodu.

Operativni sistemi

DOS: Nove verzije na pomolu

Bura koja se u poslednje vreme podigla oko poslednje verzije DOS-a polako se smiruje. Microsoft je morao da se odrekne *DoubleSpace* tehnologije – ali celoj priči tu nije kraj. Vodeći proizvođač operativnog sistema za PC računare i dva najvažnija suparnika, IBM i Novell, već najavljuju nove verzije. Kao što se i očekivalo, težište bitke za kupce prenosi se sa kompresionih tehnika na polje podrške mrežama, korisničkih interfejsa i višeprogramskog rada.

Tako Microsoft u junu planira da počne sa prodajom nove verzije koja će se verovatno zvati MS-DOS 6.3. Doneće ispravke primećenih grešaka i sopstvenu tehnologiju kompresije diskova. Zatim će uslediti MS-DOS 7.0, drugim rečima *Windows 4.0* bez grafičkog korisničkog interfejsa (GUI), koji će korisnicima pružiti pravo 32-bitno multitasking jezgro. Pitanje je, međutim, da li će im ono uopšte biti potrebno?

I IBM je objavio planove o novom izdanju DOS operativnog sistema. Opet se uglavnom radi o malim popravkama bagova, a u paketu će se naći i program za kompresiju *SuperStore* kao i nova verzija Central Point-ovog programa *RAMBoost*. Sledeća puna verzija PC DOS-a će sadržati *Workplace Shell* (Paper Software Inc.), ČMDNMĆ korisnički interfejs operativnog sistema OS/2. No, taj interfejs je očekivan još krajem 1993, ali se prema najnovijim vestima iz IBM-a na tržištu može očekivati tek 1995. godine.

Na ove najave odgovorio je i Novell, koji i dalje radi na usavršavanju Novell DOS-a, proizvoda koji je dobio kupovinom firme Digital Research Inc. Novell DOS se u sedmoj verziji pojavio u februaru ove godine, a u paketu su *Stacker* kompresione tehnologije, *pre-emptive* multitasking, *peer-to-peer* mrežne sposobnosti, *Universal NetWare Client*, kao i usluge DOS zaštićenog režima.

Operativni sistemi

Windows 3.11: Podrška za istočne Evrope

Microsoft je napokon na tržište pustio *Windows for Workgroups 3.11* koji je prilagođen korisnicima u Istočnoj i Centralnoj Evropi. Kao i kod jednokorisničke verzije, zadržane su sve osobine anglosaksonske verzije uz dodatak podrške ćiriličnim i latiničnim pismima koja se u ovom delu sveta koriste.

Microsoft ističe da je poboljšanje performansi u ovoj verziji oko 100 posto, a akcenat je stavljen i na podršku prenosnim verzijama PC računara. Tako *Windows 3.11 CEE* predstavlja kompleksno rešenje svih problema koji se javljaju u radu na našim prostorima. Takođe, Microsoft navodi da ovaj paket podržava, pored *peer-to-peer* povezivanja računara, i *Novell NetWare* lokalne mreže.

Da bi koristili *Windows for Workgroups CEE 3.11* potreban vam je barem i386SX procesor, 3 MB operativne memorije, oko 10 MB slobodnog prostora na disku, DOS u verziji 3.3 ili novijoj.

Operativni sistemi

Chicago: upravljanje sistemom

Microsoft obećava da će *Chicago* imati ugrađenu sposobnost upravljanja sistemom, ali je pitanje kada će ono biti ispunjeno. Najavljeno je da će *Windows 4.0* ove sposobnosti biti omogućene na dva načina. Prvo, preko programa za upravljanje sistemima *Hermes*, koji se očekuje „uskoro", kao i preko baze podataka *Windows Registry* i srednjih komponenti koje će biti ugrađene u buduće verzije Windows-a i Windows-a NT.

Ali, *Hermes* kasni. Prvobitno je bio zamišljeno da se pojavi zajedno sa Windows-om NT 3.1, ali se nedavno čulo iz Microsofta da će to biti tek sa NT-om 3.5, koji se razvija pod radnom šifrom *Daytona*, a očekuje se u trećem kvartalu ove godine.

Microsoft je takođe obećao da će u *Windows Registry* biti ugrađena podrška za sve upravljačka standarda, što će biti prilično teško ispuniti. *Windows Registry* je strukturirana baza podataka koja sadrži informacije o konfiguraciji i statusu svih hardverskih i softverskih komponenti određenog sistema. Tu su informacije o proizvođaču uređaja, njegova identifikacija, resursi, konfiguracija, a sva to prati i *Chicago Plug and Play*. *Windows Registry* ima i podatke o proizvođačima aplikacija, softverskim paketima, broju verzije, parametrima operativnog sistema i korisnikovoj konfiguraciji.

Windows 4.0 će imati i podršku za SNMP preko ugrađenog SNMP agenta, kao i *Management Information Base 2* hendler. Ideja o registru je veoma zanimljiva i praktična, jer osim upravljanja resursima, omogućava da se u svakom trenutku zna kome je potrebna dogradnja ili nova verzija, a ima još više smisla ako je dostupan preko WAN mreža, inače upravljanja distribuiranim računarskim okruženjem može biti vrlo teško. Tako da razvoj alatki za upravljanje sistemima uskoro može postati značajna grana softverske industrije.

HARDVER

Štampači

Canon: Još jedan konkurent

Ime Canon dugo je prisutno u svetu laserskih štampača, pre svega zbog kvalitetne mehanike koju već niz godina koristi Hewlett Packard. Nedavno je ova kompanija promovisala svoj novi uređaj koji po karakteristikama konkuriše čuvenom *HP LaserJet 4*.

Canon LBP-860 je printer rezolucije 600 tačaka po inču i deklarisane brzine od 8 stranica u minuti. Preporučena cena u Americi je 1 839 dolara, ali prema našim saznanjima u prodavnicama košta oko 1 400 dolara, što ga čini privlačnim [illegible]



Već na prvi pogled se vidi da *LaserJet 4* i *LBP-860* imaju mnogo toga zajedničkog, ali u jednoj stvari Canonov proizvod ima prednost. Ako vas interesuje *PostScript* tehnologija uštede memorije koju koristi Canon omogućava da se sa ovim jezikom za opis strane radi i samo sa 2 MB memorije u štampaču. Hewlett Packardov model za takvu opciju traži proširenje memorije na barem 6 MB. Naravno, dve megabajta nisu dovoljna za najkompleksnije poslove, tada je memoriju potrebno dograditi do najmanje 4 MB.

Kontakt adresa: Canon Computer Sistems Inc. tel 800/848-4123

Masovne memorije

Kako snimiti na CD?

Kada je pre nekoliko godina prvi put predstavljen kompakt disk kao medij na kome se mogu pohranjivati i velike količine podataka, kao jedan od značajnih adute pominjani su i uređaji za snimanje i reprodukciju. No, sve do nedavno ovakve sprave bile su prilično skupe tako da se malo ko odlučivao da arhive pravi na kompakt diskovima. Izgleda da se sada stvari menjaju.



[illegible] javnosti. To [illegible] yWrite 100 [illegible] je cena [illegible] skupo [illegible] mesec [illegible] 00 dola[illegible]

[illegible] dvostrukom brzinom na kompakt [illegible] 580 i 680 megabajta, a njihova [illegible] 30 dolara po komadu. Sve u svemu, CD-R uređaji postaju pristupačni i individualnim korisnicima koji često arhiviraju velike količine podataka.

Standardi

Univerzalne kartice

IBM u saradnji sa još tri kompanije nastoji da obezbedi veću fleksibilnost korisnicima i proizvođačima dodatnih kartica za *notebook* i desktop računare. Cilj je da se među proizvođačima PC sistema promoviše standard za tzv. SFF (*small form factor*) kartice koji je već prihvatila PCI (*Peripheral Component Interconnect*) komisija, što bi omogućilo jednostavno korišćenje i menjanje kartica različitih proizvođača. Tako bi korisnici mogli lakše da dograđuju i konfigurišu svoje računare i pri tome budu sigurni u potpunu kompatibilnost. Proizvođači će moći da koriste istu karticu kao standard i za notebook i za desktop računare, što će biti istinska *plug and play* tehnologija koja donosi i velike uštede u proizvodnji.

Western Digital je već prikazao preliminarnu verziju svoje SFF PCI Paradise video-kartice i 16-bitne Paradise zvučne kartice, a i drugi proizvođači najavljuju da su u pripremi kartice projektovane prema novom standardu.

SFF specifikacija bi omogućila projektovanje veoma kompaktnog PC sistema i pre pojave PCMCIA 3.0 CardBus specifikacije, a uz to bi predstavljala i jeftinije rešenje.

DOMAĆA SCENA

Promocije

DEC ALPHA: Prvi put među nama

„Nije prošlo ni dve mikro sekunde od trenutka kada mi je predloženo da pođem na duži put da bih pogledao jedan od najnovijih DEC računara sa famoznim Alpha čipom, pa do trenutka kada sam, kao iz topa, rekao 'Idem'. Do tada, prema raspoloživim informacijama, nije bilo ni jedne takve instalacije u najnovijoj Jugoslaviji. Šta mi je drugo preostalo?"

Stajao sam ispred računara *DEC 2000 Model 300 AXP* i čudio se kolika snaga ima u tom malom kućištu. Bio sam sumnjičav sve dok nisam bacio pogled u unutrašnjost i video za sada *DECchip 21064*, ili popularnije *Alpha* čip. Sve je izgledalo kao obična 486 mašina smeštena u nešto malo veće kućište od babi tauera sa dodatkom CD ROM, 3.5-inčnog flopi drajva i posle desetak minuta razgledanja prihvatih činjenice onakve kakve jesu.

Računar *DEC 2000 Model 300 AXP* je najslabiji model iz familije AXP računara sa ugrađenim mikroprocesorom 21064. Smešten je u kućište visine 47 cm, širine 18 cm i dubine 41 cm. Podržava do 128 MB operativne memorije, a je 32 MB minimum. Učestanost na kojoj radi CPU je 150MHz, a veličina keš memorije je 512 KB. Na matičnoj ploči postoji 6 EISA slotova u koje se može ubaciti *Ethernet* adapter, SCSI kartica sa kontrolerom za diskete kao i grafički adapter rezolucije 1024x768 tačaka. U osnovnoj konfiguraciji se dobija još i flopi uređaj 3.5 inča sa kapacitetom 2.88MB, 2 asinhrona EIA 232 porta, 1 paralelni port kao i deo za napajanje sa automatskim prilagođenjem mrežnom naponu. Na SCSI kontroler se može priključiti bilo koji SCSI uređaj, počevši od hard diskova pa do SCSI DAT jedinica, sve ukupno do 7 uređaja. Ineće konfiguracija koju sam imao ispred sebe sadržala je ADAPTEC 1742 SCSI karticu sa CD ROM drajvom i dva hard diska od 1.05 GB. Kapacitet operativne memorije bio je 32 MB. Na jednom od dva diska instaliran je operativni sistem *OpenVMS AXP V1.5* a na drugom *DEC OSF/1 AXP*. Moguće je dodati još jedan SCSI kontroler ako se radi pod *OpenVMS* operativnim sistemom, dok se uz *OSF 1* mogu dodati još tri takva kontrolera. U kućištu ima mesta za 4 uređaja (na primer CD, DAT, QIC, disk), dok se za ostale uređaje mora nabaviti dodatno kućište.

Sistem podržava dva tipa memorijskih modula: 16 MB (4x4 MB SIMM) i 64 MB (4x16 MB SIMM), oba sa vremenom pristupa od 70ns. Moguće je kombinovati ta dva tipa, pa se tako dobija sistem sa 80 MB operativne memorije. Ako se sistem koristi pod operativnim sistemom *OSF 1* može se ugraditi SVGA video adapter rezolucije 1024x768 ili video adapter visoke rezolucije 1280x1024 uz odgovarajući monitor. Naš sistem je imao instaliran SVGA monitor i adapter.

Od softverske podrške, pored operativnog sistema, može se instalirati gotovo ceo proizvodni program DEC-a. Samo ako se kupe licence, naravno. Proizvođač Digital Equipment Corporation (ili DEC) se potrudio i „prepevao" veliki deo softvera za VAX arhitekturu na novu RISC arhitekturu. Čitav asortiman uslužnih programa i prevodilaca (*DECnet, ALL IN 1, Disk Shadowing, POLYCENTER, ADA, FORTRAN, Pascal, C, C++, COBOL* itd.) se dobija na kompakt diskovima, a instalacija tog softvera je moguća jedino uz legalno kupljenu licencu. Ono što je vrlo zanimljivo jeste to da se uz operativni sistem ranije dobijalo pravo korišćenja prevodioca za asembler *MACRO 32*, dok se sada nova verzija *MACRO-64* mora kupiti.

Prema rečima proizvođača, sav korisnički aplikativni softver je prenosiv na ovu platformu, što nismo mogli proveriti. Pogodićete zašto: Nismo imali licence ni za jedan od programskih paketa (osim za operativni sistem). To je bilo naše prvo, ali i jedino razočarenje. Ali mogli smo da vidimo kako se mašina ponaša u uslovima velikog opterećenja i da instaliramo neka demo-programa koji prikazuju mogućnosti same mašine. Ovi demo programi su bazirani na grafičkim mogućnostima sistema koji se upotrebljava kao radna stanica pod *DECwindows*. Kako nismo bili u mogućnosti da utičemo na te demo programe, niti da pripremimo nekakav drugačiji test, ono što smo imali nije nas preterano uzbudilo. Od svih tih demo-programa najviše nam se dopao primer rotacije jednog antičkog torzoa u bronzi osvetljenog sa jedne strane. Ali prozor u kome se to dešavalo bio je suviše mali i nije se mogao povećati.

Što se tiče ponašanja sistema u uslovima velikog opterećenja, sproveli smo sledeći test. Kreirali smo 50 procesa koji su svaki ponaosob pravili spisak datoteka na disku u posebnu datoteku. Kada smo ih pustili da rade, mašina se malo zagrejala i u toku od 30 minuta završila svih 50 procesa. Disk je imao negde oko 50% slobodnog prostora (oko 500 MB). Takav isti test, sa manjim brojem datoteka ali istim zauzećem diska, *MicroVAX* II je odradio za 4.5 sata, to jest za oko 9 puta više vremena. Procesorsko vreme utrošeno za ovo je u istom odnosu u korist *DEC2000*. Ovaj test je rađen pod operativnim sistemom *OpenVMS AXP*. Bilo nam je jako žao što nismo bili u mogućnosti da proverimo fabričke specifikacije o performansama ove mašine u kojima stoji da je snaga mašine za *SPECint92* oko 66, za *SPECfp92* oko 113 a za *LINPACK 1000x1000* oko 90 (DP Mflops).

Ipak, mašina je ostavila pozitivan utisak i pored toga što nismo mogli da uradimo sve što smo hteli. Sa nestrpljenjem očekujemo dolazak prve *Alpha*-e u naše krajeve.

Dušan Bubalo

Korisna adresa: EuroComputer Systems, Informacioni inženjering i zastupstva, Obilićev venac 26, 11000 Beograd, telefon 638-341, 634 007, fax 185-286

Asocijacije

Udruženi korisnici CAD programa

U Beogradu je osnovana Asocijacija CAD korisnika nazvana CADUA (*Computer Aided Design Users Association*). Inicijativa je u prvoj fazi išla preko Sezama, a praćena je već klasičnim problemima sa poslovnim prostorijama koji su, kako kažu u Asocijaciji, malo usporili proces konsolidacije.

Asocijacija CAD (*Computer Aided Design*) korisnika je samostalna, stručna, neprofitabilna, nevladina, vanstranačka organizacija dobrovoljno udruženih članova. „Ulaznica" za Udruženje su katalog radova i liste referenci. O prijemu odlučuje Komisija za prijem u članstvo, na osnovu ocene materijala priloženih uz prijavu.

Članovi predsedništva Asocijacije CAD korisnika su Vladan Aleksić (potpredsednik), Aleksandar Dunkić, Đorđe Grujić (predsednik), Dušan Lazarević, Milan Mihajlović, Goran Tomić i Zdenko Šižgorić (sekretar). Ideja vodilja za uspostavljanje CADUA bila je po-

vezivanje i okupljanje ljudi koji se bave CAD-om bez obzira na platformu, te stvaranja mogućnosti za zajednički rad i delovanje, razmenjivanje iskustva, uspostavljanje i definisanje standarda na CAD polju, povezivanje sa Saveznim zavodom za standardizaciju i drugim srodnim i kompatibilnim institucijama i grupacijama, kao i nabavljanje specifične literature iz CAD oblasti.

Iskustvo pokazuje da ovakvo okupljanje i organizovanje svih korisnika CAD sistema i sistema za realizaciju aplikacija daje pozitivne rezultate pre svega kada je u pitanju razvoj oblasti, te kreativno i stručno unapređenje okruženja. Asocijacija CAD korisnika u Statutu definiše, pored opštih odredbi, ciljeve, zadatke, uslove za učlanjenje u udruženje, organe upravljanja i stručne službe.

Kontakt adresa: CADUA, Ulica Svetog Save 13, telefon: 011/446-0169

Vesna Ćosić

Asocijacije

YUCCA dobila IP mrežne brojeve

Jugoslovensko udruženje za računarske komunikacije (YUCCA) dobilo je zvanične Internet adrese (*IP network numbers*) za svoje računare. Značaj ove činjenice lako se uočava iz sledećih podataka: YUCCA je prva neuniverzitetska institucija u Jugoslaviji koja je od administratora Interneta za Evropu dobila IP brojeve, a sami IP mrežni brojevi su jedinstveni za celu zemaljsku kuglu. Tačnije, sav saobraćaj na Internetu, najvećoj svetskoj mreži koja objedinjuje dva miliona računara, odvija se upravo preko ovih jedinstvenih Internet adresa.

„Od izuzetnog je značaja to što smo zvanično registrovani u Internet bazama podataka. Sankcije UN protiv Jugoslavije se ne odnose na komunikacije i ovo je najbolji dokaz za to. Ostaje nam da i dalje radimo na povezivanju unutar zemlje, ali i povezivanju sa svetom, što je izuzetno osetljiv posao jer zavisi od dobre volje nadležnih sa druge strane veze" – obavestio nas je Radivoje Zonjić, predsednik YUCCA.

YUCCA nastavlja kontakte sa evropskim partnerima na omogućavanju razmene elektronske pošte sa svetom, kao i sa svima koji su zainteresovani za širenja računarskih mreža u zemlji.

TAKMIČENJA IZ PROGRAMIRANJA

Nova trka za mlade programere

Počelo je ovogodišnje nadmetanje nove generacije mladih programera: 2. aprila u Beogradu je održano Republičko takmičenje iz programiranja, u kojem su učestvovala 52 srednjoškolca, od kojih se njih 16 plasiralo za Savezno takmičenje. Prvu nagradu je osvojio **Janko Radusinović** (Matematička gimnazija, Beograd), drugo mesto dele **Miloš Dermanović** i **Marko Savić** (Matematička gimnazija, Beograd), a treće **Marko Milošević** (Niška gimnazija) i **Vladimir Čeperković** (Matematička gimnazija, Beograd).

Savezno takmičenje će se održati od 8. do 10. maja u Petnici kod Valjeva. Osim šesnaest predstavnika Srbije, učestvovaće pobednici takmičenja u Crnoj Gori, Vojvodini, te učenici iz Republike Srpske i Srpske Krajine.

Pred samo Savezno takmičenje, 7. maja, na poziv Društva matematičara i fizičara u Beograd dolazi reprezentacija Moskovske matematičke gimnazije pri Univerzitetu Lomonosov, koja će se sa učenicima Beogradske matematičke gimnazije nadmetati u matematici, fizici i programiranju. U decembru su naši predstavnici bili u sličnoj poseti u Moskvi.

Krajem maja mladim programerima predstoji još jedno odmeravanje snaga. Naime, od Ministarstva obrazovanja Rumunije stigao je poziv na takmičenje programera srednje-evropskih zemalja, koje će se od 26. do 31. maja održati u rumunskom mestu Kluž. Osim Rumuna i Jugoslovena, očekuju se predstavnici Austrije, Mađarske, Poljske, Slovačke, Češke, Hrvatske i Slovenije. Specijalni gosti su programeri iz Argentine i republike Moldove, kao i svi nagrađeni programeri sa poslednje programarske olimpijade u Argentini.

Pripremio: Milan Bašić



DOMAĆA SCENA

Akcije

Susret proizvođača i vlade

U organizaciji Redakcije časopisa „Računari" 25 aprila u BIGZ-u su se sreli predstavnici nedavno formirane Grupacije proizvođača računarske i informatičke opreme sa gospodinom Svetozarem Krstićem potpredsednikom Vlade Srbije U članstvo Grupacije do sad je primljeno 18 preduzeća a spisak prijava koje će uskoro biti verifikovane je mnogo duži

U ime Grupacije sastanku su prisustvovali dr Milan Kovačević (predsednik i direktor preduzeća Imtel Computers) Željko Tomić (potpredsednik i direktor preduzeća OSA – računarski inženjering) Aleksandar Mutavdžić (direktor Informatike d d) i Danko Jevtović (direktor Jugodete) sa svojim gostima Slobodanom Kokerom iz Unimatike i Nikolom Markovićem direktorom Saveznog zavoda za informatiku

U Zelenoj sali BIGZ a pred većom grupom novinara goste su u ime domaćina, pozdravili Vesna Jeremić komercijalni urednik časopisa „Računara" i Ilija Rapaić generalni direktor BIGZ-a

„Grupacija proizvođača računara i informatičke opreme nastala je u okviru Privredne komore Jugoslavije sa namerom da svesrdno radi na unapređenju računarstva u zemlji Poslednji podaci kazuju da je za ove poslove kod nas registrovano preko sedamsto preduzeća ali tek neznatni broj od njih se zaista bavi ovim poslom Stoga smo kod formiranja Grupacije odredili i kriterijume za članstvo na osnovu kojih pokušavamo da okupimo preduzeća čija je primarna delatnost informatika " istakao je dr Kovačević u uvodnoj reči

Želeći da upoznaju potpredsednika vlade gospodina Krstića sa trenutnim stanjem u ovoj oblasti predstavnici Grupacije istakli su veliki značaj povlašćenog položaja u domenu poreske politike za razvoj informatike



Gospodin Svetozar Krstić, potpredsednik Vlade Srbije



U razgovoru sa proizvođačima

„Sve do stupanja na snagu novog Programa ekonomskog oporavka računari i informatička oprema imali su status „strateškog" proizvoda odnosno smatralo se da je za ukupni razvoj našeg društva veoma važno brzo i relativno bezbolno uvođenje računara, kako u preduzeća tako i u domove Na žalost sada je to stanje donekle poremećeno jer se računari i oprema osim kada se nabavljaju kao investiciono ulaganje tretiraju sa poreskom stopom od 25 plus 2 posto " istakao je gospodin Tomić

Čulo se još da se ovakvom poreskom politikom anuliraju svi napori proizvođača da snize cene proizvoda Naime malo je poznato da je osnovna cena računara i informatičke opreme kod nas niža nego na svetskoj pijaci Kako se radi o visokoj tehnologiji sa velikim učešćem uvoznih komponenti ušteda se ostvaruje na niskoj ceni intelektualnog rada Ali kada se tome dodaju porezi stvar se menja i domaći proizvod cenom preskače granicu vrednosti u svetu

„Ovakvo stanje ostavlja dovoljno prostora za delovanje crnog računarskog tržišta i piratizovanje softvera " napomenula je Vesna Jeremić

Potpredsednik Vlade Srbije Svetozar Krstić rekao je da uvođenje informatičkih tehnologija predstavlja važnu kariku u lancu koji treba da omogući održanje koraka sa svetom a to je naročito važno sada kada su komunikacije otežane zbog sankcija Takav stav je jasno definisan u Vladi i u narednih mesec dana može se očekivati da će se poreska politika u celosti preispitati i da će se naći mogućnosti da poreska stopa za računare i informatičku opremu bude 7 plus 2 posto

„Bilo bi dobro da Grupacija svoje zahteve prosledi i nadležnim ministarstvima jer Vlada mora poštovati određenu proceduru u donošenju ovakvih odluka " istakao je gospodin Krstić

Kako saznajemo to je u međuvremenu učinjeno Zvanični predložak Grupacije upućen je na adrese Ministarstva za finansije Ministarstva za tehnološki razvoj i Ministarstva za nauku i tehnologiju

MAJSKA NAGRADNA IGRA: PETI PUT PA PENTIUM!!!

POKLANJAMO 50.000 ND U HARDVERU I SOFTVERU

POTRAŽITE SVOJU ŠANSU ZA DOBITAK JEDNE OD 80 VREDNIH NAGRADA

VELIKA NAGRADNA IGRA U ČETIRI BROJA. PREKO 70 NAGRADA I 10 PREMIJA VREDNIH OD 1000—10.000 DINARA. NAGRADNI KUPON U SVAKOM BROJU. KUPONI SE ŠALJU NA KRAJU IGRE. JAVNO IZVLAČENJE U OKTOBRU. URUČENJE NAGRADA ODMAH PO IZVLAČENJU. REZULTATI IGRE U „RAČUNARIMA 105"

Časopis „Računari", u saradnji sa grupom vodećih računarskih firmi u zemlji, pokreće novu nagradnu igru, koja se po koncepciji neće puno razlikovati od dosadašnjih — u četiri broja objavićemo četiri kupona sa vrlo jednostavnim pitanjima. Odgovore na pitanja ćete, kao i do sada, naći pažljivo čitajući tekst koji prati nagradnu igru. Četiri kupona sa četiri tačna odgovora, poslaćete, kada za to dođe vreme, na našu adresu.

Igra traje kroz četiri broja:

- računari 100 — kupon 1
- računari 101 — kupon 2
- računari 102 — kupon 3
- računari 103 — kupon 4 i kupon za adresu
- računari 105 — rezultati izvlačenja

NAGRADE

SUPER PREMIJA

1. JITEX — PENTIUM PC računar: matična ploča 4DVP—256 VLB PENTIUM 60 MHz, 16 MB RAM-a, grafička kartica VLB CIRRUS 5428, VLB IDE I/O A7CI, WESTERN DIGITAL HD 210 MB, flopi drajv 3,5", flopi drajv 5,25". U big-tower kućištu, sa 14" kolor SVGA monitorom, tastaturom i mišem

GLAVNE PREMIJE

I NAGRADA

2. OTC COMPUTER — notebook računar 486DX/33 MHz, 4MB RAM-a, VGA displej, HDD 120 MB, ugrađen track-ball, težina 3 kg

II NAGRADA

3. ADACOM RAČUNARI — računar 486DX/40 MHz VLB, 4 MB RAM-a, grafička kartica VLB 1 MB/combi VLB, HDD 215 MB, flopi drajv 3,5", flopi drajv 5,25", mini tower kućište, 14" PHILIPS kolor monitor, tastatura i miš

III NAGRADA

4. BIOSFERA — računar 386/40 MHz, 2 MB RAM-a, 170 MB HDD + štampač EPSON LQ-100 + PANASONIC telefon

PROPOZICIJE

1. Pravo učešća i mogućnost dobijanja neke od nagrada imaju svi čitaoci „Računara". U nagradnoj igri ne mogu da učestvuju lica zaposlena u firmama sponzora i u „Računarima", odnosno BIGZ-u, članovi njihovih porodica, kao ni lica koja su redovni saradnici organizatora igre.
2. Nagradni kuponi objavljuju se u četiri broja u „Računarima" broj 100, 101, 102 i 103.
3. Učesnici treba da pošalju sva četiri nagradna kupona zajedno. Za dodelu nagrada u obzir će se uzimati samo originalni nagradni kuponi iz „Računara".
4. Sva četiri nagradna kupona treba — isključivo zajedno — dostaviti na adresu „Računari", Bulevar vojvode Mišića 17, 11000 Beograd sa obaveznom naznakom NAGRADNA IGRA. Rok za dostavu 30 dana po izlasku poslednjeg kupona (u važava se poštanski žig).
5. Kuponi mogu da se šalju isključivo u standardnim kovertima manjeg formata (12,5 cm×17,5 cm). U obzir za dodelu nagrada neće se uzimati veliki koverti (formata A—5 ili veći), koverti pravljeni u vlastitoj režiji, paketići i slično. U kovertima mogu da se nalaze isključivo nagradni kuponi (4) i kupon za adresu koji će biti objavljen u „Računarima 103".
6. Redakcija „Računara" i sponzori, kao organizatori, ne mogu da snose nikakvu odgovornost za eventualni gubitak, oštećenje ili kašnjenje vašeg pisma u pošti.
7. Izvlačenja imena dobitnika nagrada izvršiće se javno. O tačnom datumu i mestu izvlačenja učesnike ćemo obavestiti preko sistema za modemske komunikacije Sezam i sredstava javnog informisanja.
8. Nagrade će biti uručene dobitnicima odmah po izvlačenju.
9. Kompletni rezultati izvlačenja će biti objavljeni u „Računarima 105".

NAGRADNI KUPON 1

računari, 100

Kako je nastalo ime JITEX?

Odgovor ____________________

POSEBNA NAGRADA

5. OSA — softvarski paket AutoDesk AutoCAD 12 – diskete, dokumentacija i CD ROM

PREMIJE

6. JUGODATA — hardver ili softver po izboru dobitnika u vrednosti od 1500 bodova

7. LS DATA — džepni računar PSION series 3, 256 KB RAM-a

8. INTERSOFT — štampač EPSON LQ-570

OSTALE NAGRADE

9. AB SOFT — štampač EPSON LX-400 i softver za računovodstvo i knjigovodstvo (dve nagrade)

10. PREDUZEĆE MZ — štampač EPSON STYLES-800 ink-jet 360*360 dpi, automatski dodavač papira

11. COMPUTER BEOGRAD — štampač EPSON LQ-1000

12. SAGA — štampač STAR LC-15

13. MICROAIR — Video konverter VGA-TV Encoder

14. CET, MISOFT i SOFTLAND — softverski paketi QuattroPro for Windows i Workgroup Add-on for Window 3.11, knjige, diskete i kalkulator (13 nagrade)

15. ASYS COMERCE — štampač EPSON LQ-100

16. MICROSYS — štampač EPSON LQ-100

17. VF d.d. — profesionalni modem PP 1200 GS (dve nagrade)

18. PERIHARD — kasete za lasere HP3 i HP4 (tri nagrade)

19. MICRO ANIMA — kursevi za AutoCAD i CORELDRAW (6 nagrada)

20. PROSOFT — Novell PC DOS 7.0 (1 nagrada), kutija disketa (10 nagrada)

21. APP System — knjige (4 nagrade)

22. INSTITUT „BORIS KIDRIČ" VINČA — knjige (13 nagrada)

23. RAČUNARI I SEZAM — pretplate (12 nagrada)

U POTRAZI ZA LOGIKOM

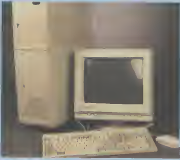
U februeru 1990 godine osnovano je preduzeće JITEX, kao plod želje grupe stručnjake de stvore mesto na kome mogu de realizuju svoje ideje Problem imene leko je prevaziđen — JITEX je u stveri kovanica nastala spajanjem skraćenice JIT i EX koje označavaju **Just In Time,** operacioni metod poreklom iz Japana i ekspertne sisteme Danes su osnovne oblasti rada ove kompanije razvoj softvera, neročito u oblasti upravljanja proizvodnjom odnosno proizvodnja PC računara i industrijske elektronike

— Nepunih godinu dana pre osnivenje JITEX-e počeli smo red na istraživanju problema logičkih struktura U fazi razvoje softvere ove namene keo odlican poligon za testiranje izdvojile su se šehovske partije i kako to obično biva iz jedne ideje rodila se druga — izrada programa koji bi unete šehovske partije obradene kao logičke strukture pripremio za štempanje Naravno morali smo obim šehovskih partija dovesti do nivoa koji je pngoden za jednu knjigu Konečna odluka bila je obrede partija Gnnfildove indijske odbrane — upozneo nas je sa počecima JITEX-a direktor, gospodin Radmilo Bujdic

U ovom preduzeću realizovani su mnogi interesantni projekti Dugogodišnji rad u domenu razvoje terminela osetljivih na dodir (TID terminali) rezultirao je izradom programa GET-C To je generetor izvornog koda za upravljanje radom TID terminela koji programeri kasnije ugrađuju u sopstvene aplikacije

Kada govonmo o razvoju softvere u preduzeću JITEX, neizbežne tema je KUP-M To je peket ze logističku podršku proizvodnji koji čine 32 programa kao i više operacionih metode Glevni produkcioni plan glavno planiranje resursa i kontrole produkcionih aktivnosti čine osnovu ne kojoj se zesniva KUP-M Nerevno za ovako složen softverski sistem obezbeđene je i hardverska podrška na PC pletformi Deo potrebnih informecija prikuplja se preko TID terminale, bar—kod čiteča i mernih instrumenata KUP-M je razvijan dve godine prvu promociju imao je ne prošlogodišnjem Sajmu tehnike u Beogredu, a u komercijelnu eksploataciju usao je u decembru prošle godine

Ideje osnivača preduzeća JITEX — istraživanje u oblasti logike i primena u svakodnevnom životu pokazala se ze ove četiri godine keo pravi put uspeha Priču o potrezi za logikom preduzeća JITEX nastavljamo u sledećem broju



JITEX, Beograd Oblakovska 57/a
tel (011) 650-777 tel/fax (011) 617 141



OTC, Beograd, Milentija Popovića 9 SAVA CENTAR
tel (011) 2222 355



ADA COM Beograd Čika Ljubina 12
tel (011) 629-233 337 367 344-492



BIOSFERA, Beograd Makedonska 22
tel/fax (011) 3225-109 3224-378 3248 208



OSA, Beograd Narodnog Fronta 56
tel (011) 681 178 681 199 fax (011) 681-743

jugodata

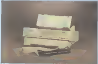
1500 DIN

JUGODATA, Beograd Bulevar Revolucije 326
tel/fax (011) 418-326 418-768

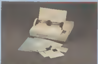


LS DATA, Beograd Gračanička 8/1
tel (011) 626-758 628-056/18

SVE OVO JE — VAŠE



INTERSOFT Beograd 29 novembra 43/I
tel (011) 322 1959 322-3539 322 5393 322-4120



AB SOFT, Beograd Kneza Miloša 82
tel (011) 644 255/113 128 210 fax (011) 656 857



MZ PREDUZEĆE, Beograd, Dubljanska 70
tel (011) 434-812 fax 450-471



COMPUTER BEOGRAD Beograd I Suljeska 19a
tel (011) 711 916 fax (011) 711-478



SAGA, Beograd M Popovića 9 SAVA CENTAR
tel (011) 222-3678 147 182 222 4323 ext 256



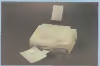
MICRO AIR, Beograd, 29 novembra 71
tel (011) 343-294 tel/fax (011) 343-840



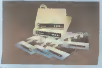
CET MISOFT I SOFTLAND Beograd Skadarska 45
tel/fax (011) 343043



ASYS COMERCE Novi Sad Bulevar oslobođenja 58
tel (021) 623-928 624-501 616-867



MICROSYS Beograd Molerova 70 Beočin Novi Sad
tel (011) 432-680 430 059 tel (021) 611 366 52 550



FVDD, Beograd Cvetna 4
tel (011) 612-211 611-611 fax (011) 108-454



PERIHARD, Beograd Ivana Milutinovića 24
tel (011)(436-019



MICRO ANIMA, Beograd 29 novembra 71
tel (011) 343 294 tel/fax (011) 343-840



PROSOFT Kragujevac Ul JNA 63
tel/fax (034) 212-806



ATP SYSTEM, Beograd Jovana Ristića 6
tel/fax (011) 647 190



INSTITUT „BORIS KIDRIĆ Beograd Nemanjina 4 1C
tel (011) 683 380 682-486 641 155 107 181



NASAMO SA ALFOM

Nedavno nam se pružila prilika da pogledamo, sasvim izbliza, dve možda najzanimljivije mašine u PC liniji firme Digital – DECpc sa Alfa i Pentium procesorom. Pokazalo se, međutim, da je prenosivost softvara dosta uslovna stvar i da uobičajena test alatka neće da rade na obe platforme. Preostaje nam, dakle, samo da vam prenesemo kakva je i na šta liči mašina koju njen proizvođač skromno naziva „najbrži i najmoćniji PC na planeti".

Jovan Regasek

Ovim reklamnim sloganom Digital zapravo otkriva svoje namere ne samo da prodre na tržište personalne informatike, nego i da njime potpuno ovlada. On, međutim, nije za taj prodor obavio ni blizu tako temeljite pripreme kao što su to učinili Apple, IBM i Motorola sa procesorom PowerPC. Zbog toga DECpc AXP 150, iako se nalazi na tržištu već gotovo godinu dana, prolazi potpuno nezapaženo. Obećanje da će se do kraja ove godine pojaviti 1000 namenskih 32-bitnih aplikacija za Alfu, koje je Digital dao na promociji mašine, teško da može biti ispunjeno, jer se njihov broj još uvek može izbrojati na prstima jedne ruke. Digital se, međutim, ne uzbuđuje previše. Stoji i posmatra sa strane ljutu bitku između procesora PowerPC i Pentium, prepuštajući konkurenciji da navikne tržište na RISC tehnologiju i utre put za pobednika. Sa bez sumnje najjačim RISC procesorom za PC klasu računara, lako se može dogoditi da to na kraju ipak bude – Digital.

PC ILI RADNA STANICA

U računar DECpc Alfa je ugrađen DECchip 21064, 64-bitni RISC (sam Digital ga naziva „superskalarni procesor sa super linijom za instrukcije") na 3,3 V, sa unutrašnjim radnim taktom na 150 i spoljašnjim na 50 MHz. Ovaj procesor (o kome čitaoci „Računara" sigurno već znaju sve) analitičari smatraju jednim od najjačih RISC proizvoda PC klase na tržištu, ali sam Digital, nakon što je lansirao model na 275 MHz, počinje da govori o njemu pomalo i u prošlom vremenu. Procesorski keš na samom čipu (16 KB) je ojačan drugostepenim kešom kapaciteta 512 KB. Alfa je projektovana u sada već pomalo arhaičnoj EISA arhitekturi.

Osnovni model ima 32 MB radne memorije (proširivo do 128 MB), mali flopi od 2,88 MB, SCSI tvrdi disk od 425 MB, EISA Compaq QVision grafičku karticu sa maksimalnom rezolucijom 1280x1024, EISA Ethernet mrežnu karticu i CD ROM jedinicu od 600 MB, a dolazi sa Sony kolor monitorom od 17 inča. Tako u najkraćim crtama izgleda mašina stvorena sa neskrivenim ambicijama da postane najači PC na svetu, ali i da radi kao radna stanica na najtežim poslovima u CAD-u, stonom izdavaštvu i multimediji. Iako se Digital trudi da u svakoj prilici istakne PC komponentu, DECpc AXP i po svojim performansama i po svojoj ceni (od 6795 dolara naviše) ipak pre spada u svu drugu kategoriju.

DECpc Alfa je, kažu, RISC procesor uklopljen u intelovu arhitekturu, ali to po matičnoj ploči nikada ne biste rekli. U pitanju je najsloženija matična ploča koju je autor ovih redova ikada video u jednom PC računaru, sa samo dva prepoznatljiva detalja: EISA slotovima i SIMM podnožjima sa 72 pina. Ploča je rađena u takozvanoj „diskretnoj tehnici", dakle sa logičkim kolima umesto sa visokointegrisanim čipovima, i u poređenju sa jednostavnim i elegantnim Pentium pločama izgleda kao da dolazi iz drugog sveta. Svima je očigledno, dakle, da je u pitanju neka druga tehnologija. Još jedan detalj na ploči je privukao našu pažnju, podsetivši nas na pregrejana izjave g. Pole Kmate (u intervjuu koji smo svojevremeno objavili u „Računarima") o „hladnoj Alfi" i „vrelom Pentiumu". Tako ogromne hladnjake (hladnjak na Pentiumu je prava igračka!) smo imali prilike da vidimo samo u audio pojačivačima snage 50 W i više! To, naravno, ne čudi kada se zna da Alfa oslobađa 28 W toplote!

UNIVERZALNA PLATFORMA

DECchip 21064 je projektovan kao univerzalna platforma i na njemu se može izvršavati više operativnih sistema, bez potrebe da se kod sa drugih platformi u potpunosti prilagođava novom procesoru. Uz uobičajena tri tipa instrukcija (operacije, upravljanje memorijom i grananje) Alfa ima i četvrti koji, pod imenom PALcode (*Privileged Architecture Library Code*) obuhvata grupu rutina za izvršavanje instrukcije koje su jedinstvena za pojedine operativna sistema. DECpc Alfa dolazi sa operativnim sistemom *Windows NT*, a može se podići i kao Open VMS ili OSF1/UNIX konzola.

Windows NT je izabran iz nekoliko razloga. Između ostalog i da obezbedi Alfi vezu sa ogromnom bazom DOS i *Windows* aplikacija. Te aplikacije se, naravno, ne izvršavaju preko 32-bitnog *NT* koda (Digital tvrdi da će se *NT* aplikacije izvršavati dva puta brže na Alfa nego na Pentium platformi), nego preko emulatora Intel procesora, koji je razvila firma Insignia Solutions. Kod emulacije procesora su bitne dve stvari: brzina i kompatibilnost, i otuda su se te dve komponente našle u centru naših istraživanja.

Emulacija Intelovih procesora, zbog njihove glomazne i komplikovane arhitekture, nije nimalo jednostavna stvar, ali se Alfa u tome, kad je reč o brzini, snalazi izuzetno dobro. Imali smo utisak sve vreme da se pred nama nalazi 486 mašina najviše klase, sa posebno blistavim trenucima u poslovima koji se uglavnom oslanjaju na procesor (kao što je popunjavanje površine komplikovanim teksturama u programu *CorelDraw*). Prema navodima Digitala, Alfa u emulaciji razvija snagu 486DX mašine na 25 MHz, dok je prema rezultatima koji su nama stajali na raspolaganju iz literature, brzina Alfe sa DOS i Windows softverom znatno manja i iznosi samo desetinu brzine Pentium računara. Bez obzira, međutim, na ove brojke koje mogu biti i ovakve i onakve, brzina emulacije je zaista bila fascinantna.

NEVOLJE SA EMULACIJOM

DECpc AXP na prvi pogled funkcioniše potpuno besprekorno, kao da je rođen sa DOS i Windows softverom. *Windows NT* je, čim smo mašinu priključili u Novell mrežu, odmah prepoznao sve stanice sa daljim diskovima u sistemu. Lako smo preko mreže sa sopstvenog noda prebacili sve softverska alata koji su nam bili potrebni za testiranje mašine, startovali sa Alfe omiljeni teksi-procesor koji se i dalje nalazio na našoj stanici i još poneki program – i sve je išlo kao po loju.

Prve ozbiljnija probe su, međutim, pokazale da je prenošenje operativnih sistema sa jedne procesorske platforme na drugu i izvršavanje „tuđih" aplikacija više nego osetljivo pitanje. *CorelDraw* je radio savršeno, a *PhotoPaint* iako iz istog paketa, nije mogao ni da se pokrene. *AdobePhotoshop* je odbio da se instalira jer „radi samo sa procesorima 386 i višim". *QuarkXpress* takođe, ali iz drugih razloga. Alatka za merenje ICOMP performansi je pronašla da je u pitanju 286 procesor na 5 MHz (!), a kazaljke na brzinomeru su zakucala na „impozantnih" 78 (očekivana vrednost između 800 i 1000). Alatke iz paketa *Norton Utilities* su takođe odbijala da rade.

Problem je, naravno, odmah bio jasan. Na mašini DECpc APX neće raditi ni jedna aplikacija koja ima bilo kakvog direktnog posla sa hardverom. Pretpostavljamo da će sledeće verzije DOS i *Windows* softverskih paketa biti prilagođena i ovoj okolnosti, odnosno da će sve usluge od hardvera tražiti isključivo preko operativnog sistema, ali je situacija trenutno više nego obeshrabrujuća. Na Alfa računaru sasvim sigurno neće proraditi baš svaki DOS i Windows program koji uzmete sa police! Nakon pravih bezuspešnih pokušaja sa instalacijom raznih softverskih paketa, ostajemo u prilično neprijatnoj dilemi da li Alfa uopšte emulira 386/486 klasu, ili su se u Digitalu možda zadovoljili emulacijom samo 286 procesora.

ALFA ILI PENTIUM

Na spisku softvera koji ne radi nalaze se i sve uobičajena softverska alatka koje se u PC svetu koriste za testiranje performansi. Nismo, na žalost, uspeli da dođemo ni do *Windows NT* razvojnog kita za Intel i Alfa platformu, kojim bismo rekompajlirali neophodan softver. Time je pao u vodu ne samo plan da propisno izmerimo performanse DECpc AXP mašine, nego i da uradimo realni uporedni test na relaciji DECpc Alfa / DECpc Pentium. Radio je praktično samo Winbench 4.0, koji na žalost ne obuhvata performanse procesora, ali ipak daje izvesnu predstavu o tome koliko emulacija usporava Alfa mašinu. Brzina DECpc AXP displeja je 3,18 megapiksela u sekundi, a propusna moć diska 317 KB/s, što je u oba slučaja jedva četvrtina snaga DECpc XL Pentium računara. Ne testovima se namenskim softverom, međutim, prema podacima kojima raspolažemo iz literature, Alfa je bila brža od Pentiuma ne više od 30 do 50%, što, s obzirom na razlike u arhitekturi i radnom taktu procesora, možda i nije rezultat koji obara s nogu. Kad bismo hteli da u spekulacijama idemo i dalje, mogli bismo da kažemo i to da su Alfa i PowerPC, ugrađen u PowerMac računare, prema onome kolika odmiču Pentiumu, dakle samo posredno, po snazi približno isti.

Uporedo sa DECpc AXP 150, na našem stolu je stajala i jedna DECpc XL mašina, koja se po izgledu, jer se kosrti istovetno kućište, razlikovala samo u nekoliko dizajnerskih detalja. U pitanju je tzv. „processor independent PC", univerzalna PC platforma u koju se može ugraditi bilo koji procesor, od 16-bitnog Intela do 64-bitne Alfe. Digital je, kao i Compaq svojevremeno u računarima SystemPro, premestio procesor sa matične ploče na lako izmenljivu karticu, koja se povezuje sa pločom preko specijalnog (malo je reći) mašipolnog PCI slota. DECpc XL klasa je projektovana u PCI arhitekturi sa integrisanim SCSI II interfejsom, a u našem modelu (DECpc XL566) nalazila se procesorska kartica sa Pentiumom na 66 MHz, PCI S3 928 grafička kartica, SCSII disk od 512 MB i CD ROM sa dvostrukom brzinom (i izuzetno finim mehanizmom za prihvatanje CD diska).

U pitanju je, ukratko, do savršenstva izbrušena Pentium mašina (prema testovima PCBench 8.0 brzina procesora 77,96, brzina grafičke kartice 6766, DOSMark rejting 1007, Graphics WinMark 14,2, DOS WinMark 1260 – većina rezultata bolja od svega što smo do sada imali prilika da vidimo). Zbog čega pričamo sve ovo? Zato što bismo, kao okoreli stanovnik DOS i pomalo Windows sveta, birajući između dve mašina koje su stajala jedna pored druga, između DECpc Alfa i DECpc Pentiuma, ipak izabrali ovu – drugu. Oponašanje Intel platformi je, međutim, samo usputna epizoda u razvoju mašina, dok ne pristigne neophodna podrška proizvođače softvera i hardvera. Osnovna namena DECpc AXP računara je da obavlja više poslova za više korisnika istovremeno na nekom od 64-bitnih operativnih sistema, u kojima Digital, prema sopstvenim tvrdnjama, ne oskudeva ni sada.

BEZ RISCA NEMA ZABAVE

Izgleda da reklamna krilatica iz našeg naslova (u originalu: No RISC, No Fun!), uz koju je svojevremeno lansiran jedan od prvih RISC procesora, tek ovih dana dobija svoj puni smisao. Na tržištu procesora vodi se, ovih meseci, jedna od presudnih bitaka za prevlast na PC tržištu. Na jednoj strani je Intel sa svojom CISC tehnologijom, a na drugoj čitave horde proizvođača RISC procesora, među kojima je, van svake sumnje, PowerPC u prvoj borbenoj liniji. Na ovoj sceni, dakle, ima zabave i uzbuđenja na pretek, ali u RISC zabavi, reklo bi se, ne uživaju svi podjednako. Intel, svakako, najmanje.

Jovan Regasek

RISC naravno nije tehnologija od juče. Ovi procesori u raznim varijantama, rade u brojnim radnim stanicama i velikim mašinama na najtežim računskim poslovima. Čak ni PowerPC nije doživeo svoju svetsku premijeru sredinom marta kada je Apple lansirao čitavu familiju računara sa ovim procesorom, nego nekoliko meseci ranije, daleko od očiju i ušiju javnosti, u novim radnim stanicama firme IBM. Ključni, a možda će se pokazati i sudbinski korak bio je međutim ulazak Apple PowerPC platforme na masovno tržište personalne informatike, u onaj ogromni prostor koji Microsoft i IBM već više od deset godina grade na Intelovim procesorima – od već potpuno zaboravljenog 8088 do najnovijeg Pentiuma.

STRATEGIJA: DUŠA JEDNE NOVE MAŠINE

Osnovni oslonac PowerMac mašine nalazi se u ROM-u kapaciteta 300 K sa emulacionim softverom koji omogućuje da se na novom „meku" izvršava svaki program koji je ikada napisan za „makintoš" platformu. Deo koda je posvećen emulaciji procesora 68040LC (model bez matematičkog koprocesora – Apple je došao do zaključka da se ovde emulacija ne isplati), a deo kompletu rutina pod nazivom *Toolbox* preko kojih „makintoš" komunicira sa periferijama i gradi svoj čuveni GUI interfejs. Očekivalo bi se da je *Toolbox* pisan u prirodnom kodu za PowerPC. Ali nije. Apple je, u oskudici vremena, prearadio samo ključne rutine jer ga zapravo očekuje ogroman posao. Rutine za *Toolbox* su pisane na mašinskom jeziku za Motorolin procesor, pri čemu je osnovni kriterijum bila efikasnost izvršavanja koda. I ne mogu se lako, bez teškog rada, preneti na drugu platformu. Stoga emulator predstavlja mešavinu 680x0 i PowerPC koda. Operativni sistem *System 7.1* takođe dolazi u starom kodu, ali je Apple-ov problem ovde nešto jednostavniji jer je on pisan u pravovremenom C++ jeziku i treba ga samo – prekompajlirati.

Mešani kod će, izgleda, predstavljati uobičajenu stvar u aplikacijama za PowerMac. Za operativni sistem neće predstavljati nikakvu teškoću da u istom programu čas izvršava 680x0 a čas PowerPC kod. I softver za PowerMac će po svemu sudeći takođe dolaziti po principu „dve verzije u istom pakovanju" sa inteligentnim rutinama za instalaciju programa. Sa stanovišta korisnika između starog i novog „meka" ne zapaža se nikakva razlika osim, razume se, u snazi mašine.

Prilikom projektovanja PowerMac mašina Apple je imao tu osnovnu cilju da snagu svog legendarnog „makintoša", koji se po brzini praktično nije mogao ni porediti sa PC računarima najnovije generacije, podigne na novi stepenik, da prelazak sa jedne na drugu platformu za „mekove" korisnike bude potpuno neprimetan i bezbolan i, konačno, da iskoristi dobitak u snazi i prenese aplikacije iz DOS i *Windows* sveta na sopstvenu platformu. Snaga nove mašine je, u odnosu na ono na šta su navikli korisnici „makintoša" i „kvadri", zaista impresivna – sa PowerPC softverom PowerMac je brži i do šest puta od najbržih „kvadri". Zahvaljujući emulaciji procesora 68040 novi računar izvršava, doduše sa izvesnim usporenjem, sve aplikacije na koje su korisnici navikli na staroj platformi. Preko programa *Soft Windows*, emulatora *Windows*-a za *System 7.x*, PowerMac izvršava i sve DOS i *Windows* aplikacije. Sa izuzetno velikim naporom, kažu kao 486SX na 25MHz, ali ipak – izvršava.

Da li je to dovoljno za uspeh na tržištu?

SOFTVER: RAŠIRENIH RUKU

RISC je do sada bio marginalna procesorska tehnologija uglavnom zbog odsustva hrabrosti u softverskoj industriji da ga podrži aplikacijama za masovno tržište. Odnos prema RISC tehnologiji se, međutim, ovih dana iz temelja menja. Izgleda da svako želi da uhvati priključak sa novom klasom procesora. PowerMac je već uspeo da privuče Microsoft, WordPerfect, Aldus i Adobe. Svi oni su spremni ili će to biti u nekoliko narednih meseci, da puste na tržište namenske PowerMac verzije svojih popularnih softverskih paketa. Ovi velikani softverske industrije su, međutim, samo prva četvorka na dugačkoj listi proizvođača koji najavljuju svoju rešenost da se uključe u veliku PowerPC utakmicu. Na prvoj javnoj demonstraciji, sredinom marta u Njujorku, najavljeno je čak 150 aplikacija napisanih (da budemo precizniji: rekompajliranih) specijalno za PowerPC platformu, i njihov broj raste iz dana u dan. I ne samo to. Neki među njima, kao što je Adobe, su raspoloženi da starim kupcima (koliko starim? pa ne baš previše) isporuče besplatno PowerPC verzija svojih softverskih paketa.

Sudbina novih procesora je oduvek bila u rukama softverske industrije i zbog toga su sve oči uprte u Bila Gejtsa u očekivanju da vide na čiju će stranu softverski gigant Microsoft da stane ovoga puta. Gejts, doduše, nije ušao vremena da se pojavi na svetskoj premijeri, ali je, u javljanju uživo preko satelita, klimnuo glavom i rekao „Da!". Izgleda da će u mesecima koji dolaze Microsoft izbaciti na tržište PowerMac verzije svih svojih ključnih softverskih proizvoda, a pre svega kompleta *Microsoft Office* integrisanog paketa *Microsoft Works* i baze podataka *FoxPro*. Prvi obuhvata programa *Excel*, *PowerPoint* i *Word*, a drugi predstavlja integrisanu aplikaciju za obradu teksta, komunikacije, radne tabele i baze podataka. PowerPC verzija ovih poznatih programa će, obećavaju u Microsoftu, koristiti do poslednjeg atoma snagu PowerMac platforme.

Za PowerPC platformu je u pripremi i nekoliko ozbiljnih operativnih sistema. Uz već postojeći operativni softver koji će biti prenet sa drugih platformi – AIX (IBM), Solaris (SunSoft), *Windows NT* (Microsoft) – očekuju se i IBM-ov *Workplace OS* kao *PowerOpen* i *Taligent* koji zajednički razvijaju IBM i Apple. Pri tome svaki među njima dolazi sa ambicijom da, rekompajliranjem koda ili kroz emulatorski režim, obuhvati što više aplikacija sa drugih platformi, naročito iz DOS i *Windows* sveta. Preko PowerPC verzije *Windows* emulatora *Soft Windows* na PowerMac-u može da se pokrene bilo koja aplikacija sa Intel platforme. Solaris će imati WABI koji će direktno izvršavati *Windows* kod. Softver, očigledno, više ne predstavlja nikakav problem. Softversko blago koje je za mašine sa Intelovim procesorom stvarano godinama na PowerPC platforme će, izgleda, biti preneto preko noći. Softverska podrška, međutim, neće biti dovoljna ako novu platformu ne podrže i proizvođači hardvera, ali IBM, Motorola i Apple (tvorci PowerPC procesora) ni tu ništa ne prepuštaju slučaju.

HARDVER: REFERENTNA PLATFORMA

IBM je odlučio da i sa PowerPC procesorom ponovi formulu otvorene arhitekture na kojoj je zapravo i izrasla čitava personalna informatika. U saradnji sa Motorolom definisao je i objavio potpunu specifikaciju za PowerPC platformu pod nazivom PowerPC Reference Platform (PReP). Standard su oberučke već prihvatili veliki proizvođači matičnih ploča na Tajvanu (Mital, DTK, Tatung i UMC), ali se Motorola i IBM nisu zadovoljili samo time, već su sa odgovarajućim grupacijama formirali konzorcijum pod nazivom New PC sa zadatkom da promoviše novi standard. IBM planira da isporučuje kompletne PReP matične ploče, ali i samo procesore i kontrolnu logiku onima koji bi više voleli da ploče prave sami. PReP specifikaciju spremno prihvataju i proizvođači periferijske opreme (ATI, Cirrus, Matrox i WesternDigital) obećavajući kako softversku podršku, tako i novu generaciju čipova koja bi bila potpuno prilagođena PReP arhitekturi.

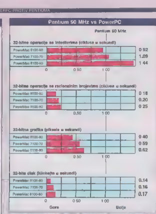
PReP specifikacija je potpuno fleksibilna. Softver i hardver ne komuniciraju direktno, nego se između njih nalazi tzv. apstraktni međusloj koji omogućuje dizajnerima da oblikuju PowerPC sisteme potpuno slobodnih ruku, unapređujući njihove performanse po svojoj volji, bez bojazni da će se ogrešiti o softversku kompatibilnost. Apstraktni sloj PowerPC mašine (MAL) se sastoji od tri segmenta. Prvi segment je blok

NAJNOVIJA INTEL GENERACIJA

Procesor	Spoljašnja brzina (MHz)	Unutrašnja brzina (MHz)	Adresni bas (bita)	Data bas (bita)	iCOMP	Keš K	FPU Integrisana	Napajanje
Pentium 100	66	100	64	64	815	16	da	3.3V
Pentium 90	60	90	64	64	735	16	da	3.3V
DX4 100	50 ili 33	100	32	32	435	16	da	3.3V
DX4 75	25	75	32	32	316	16	da	3.3V
486DX2/66	33	66	32	32	297	8	da	5V
Pentium 66	66	66	64	64	567	16	da	5V

Iz postojeće tehnologije Intel sve teže izvlači novu snagu





Još uvek izazanje: Najnoviji Pentium na 90 MHz je uspeo da dostigne tek PowerPC na 66 MHz. Možda je, međutim, to prilično realan odnos snaga jer je spoljašnji takt oba procesora u tom slučaju isti

koda za podizanje sistema koji se nalazi u ROM-u na matičnoj ploči. Njegov zadatak je da pronađe i učita PReP kompatibilan operativni sistem i potom proveri i pripremi hardver u mašini. U drugom segmentu se nalazi kod koji omogućuje operativnom sistemu logički pristup hardverskim komponentama, dakle ne vodeći računa o njihovoj fizičkoj prirodi. Ovaj kod koji se inače naziva apstraktnim slojem sistema (SAL) i koji je sličan apstraktnom sloju hardvera (HAL) u programu *Windows NT* isporučuju ili proizvođač operativnog sistema ili proizvođač mašine. Treći segment konačno predstavlja skup procedura za pogon perifernih uređaja (drajvera). IBM planira da vodećim proizvođačima hardverskih komponenti (ATI, Matrox, Hewlett-Packard) omogući da ugrade sopstveni kod u apstraktni sloj operativnog sistema *Workplace OS*.

Osnovna hardverska specifikacija PowerPC mašine je prema PReP preporuci izuzetno skromna: 8 MB RAM-a, disk od 80 MB, grafika u rezoluciji 640×480 sa 256 boja, 16-bitni stereo zvuk sa uzorkovanjem od 44.1 kHz, serijski priključak brzine 19.2 i paralelni priključak za štampač. Očekuje se, međutim, da će tipična PowerPC konfiguracija imati 32 MB RAM-a, tvrdi disk od 240 MB, flopi od 1.44 MB, CD-ROM XA jedinicu, ubrzanu 16-bitnu grafiku u rezoluciji 1024×768, stereo zvuk sa kompresijom podataka, SCSI 2 interfejs, Ethernet i veoma brze serijske i dvosmerne paralelne priključke. Matična ploča je standardnih dimenzija i može se ugraditi u svako PC kućište. Prima PowerPC procesore na 60, 66 i 80 MHz. Mesto za drugostepeni procesorski keš nije predviđeno, ali se on može ugraditi pomoću odgovarajuće kartice. Memorija se preko osam SIMM podnožja sa 72 pina može proširiti do 256 MB.

Zanimljivo je da baš arhitektura nije definisana PReP standardom, ali se na IBM matičnoj ploči nalaze obe, i PCI i ISA magistrala. Na PCI magistralu direktno su vezani fleš ROM kapaciteta 512 K sa kodom za testiranje i podizanje mašine, NCR 53C810 SCSI-2 kontroler i dve slote za proširenja, pri čemu će jedan svakako biti zauzet grafičkom karticom sa Weitek Power 9000 ili S3 928 PCI video procesorom. Preko Intelovog bridž čipa 82378IB (most između dve arhitekture) PCI je vezan na ISA magistralu.

IMPAKT: INTERFEJS ZA DEVEDESETE

Otkuda tolika spremnost softverske industrije da podrži jednu novu i očigledno nedovoljno proverenu procesorsku platformu? Možda iz osećanja da je Intelova CISC tehnologija, na čijim leđima je izrasio ono što se danas zove personalna informatika, praktično na izdisaju. Proizvođači softvera sve teže izvlače snagu iz Intelovih procesora, a skromni dobici u performansama od 50, 80 ili 100% koliko donose njegove nove generacije nisu, izgleda, ni iz daleka dovoljne da prate užurbani tempo softverske industrije, kojoj je produktivnost na prvom, a efikasnost koda na poslednjem mestu. PowerPC, čija je prva generacija po performansama u najmanju ruku poredive sa poslednjom Intelovom, u ovom času predstavlja sasvim svež izvor snage sa, reklo bi se, poprilično rezervama. Toma će malo ko hteti da odoli. I sam IBM, prema poslednjim vestima, odustaje od svojih prava da proizvodi Pentium i Intelove procesore potonje generacije. Da li to znači da će se na IBM PC računarima i njihovim kompatibilcima uskoro, umesto „Intel inside" nalaziti znak „PowerPC inside"?

PowerMac je namenjen, kaže Apple, „velikim investitorima u informacione tehnologije kojima je potrebna RISC snaga, ljudima koji se bave izdavaštvom, grafičkim oblikovanjem i multimedijom i, konačno, DOS i *Windows* korisnicima", a prenosivost sa platforme na platformu i brzina u grafičkim i multimedijalnim aplikacijama predstavljaju osnovna obeležja ove nove mašine. Veća snaga je, međutim, značila uvek više. Ona je pre svega otvarala vrata za nove ideje za primenu računara na nove načine.

Sadašnja generacija personalnih računara je u stanju da automatizuje poslove koji se odnose na obradu teksta, radne tabele i baze podataka. Ali ti poslovi „nisu najvažniji za način na koji većina ljudi koristi svoje vreme", smatra Apple. „Naš cilj je da tehnologiju približimo svim aspektima ljudskog života." Originalni „mekintoš" interfejs je bio dobar za osamdesete. Ali biznisu je, smatra Apple, potrebno mnogo više. Potreban mu je interfejs sa „govorom, glasom i gestovima". U danima koji dolaze Apple planira da potencijale nove platforme iskoristi za projektovanje novog korisničkog interfejsa pod nazivom „aktivni agent" i najavljuje aplikaciju za Internet, univerzalni *mailbox* za elektronsku poštu, faksove i glas, *OpenDoc* za prenos elektronskih dokumenata sa jedne platforme na drugu i „nešto važno u vezi sa Unix-om".

SNAGA: PowerMac JE JAČI

Stvarna snaga PowerPC mašina, međutim, još uvek nije poznata. Proizvođači softvera, naravno, uz dižu svoje PowerPC proizvode i puni su hvale u oceni PowerMac mašina. Microsoft izjavljuje da zapaža „značajne dobitke u brzini u PowerMac verzijama *Word*-a i *Excel*-a". *Photoshop* i *Fractal Painter* se prema navodima firme Adobe izvršavaju brže na PowerMac nego na Pentium mašinama. U kurtoaznim promotivnim izjavama loše se provela čak i Sun SPARCstation radna stanica koja navodno izvršava program *Mathematica* sporije od PowerMac-a! Prvi nezavisni testovi koje je obavio časopis MacWorld takođe su išli u prilog PowerMac-u. Razlike u brzini

nisu bile spektakularne do tridesetak procenata u korist nove mašina ali su ipak označile da je nešto pomolo bolje

Uporedni testovi sva tri člana PowerMac familije na jednoj i dve mašine sa Pentium procesorima na 66 i 90 MHz, na drugoj strani koje je sproveo *PC Magazin UK* pokazali su veoma iznenađujuće rezultate PowerPC je tukao obe Pentium verzije u 32-bitnim procesorskim testovima – najbrži PowerMac na 80 MHz bio je 73% brži od Pentiuma na 66 i 44% brži od Pentiuma na 90 MHz U radu sa racionalnim brojevima, s druge strane Pentium je bio više nego superioran – PowerMac nije mogao da razvije više od 31% snage najslabijeg Pentiuma Korisnici CAD paketa i uopšte softverskih alata u kojima se intenzivno koriste operacije u pokretnom zarezu sigurno neće požuriti previše da promene platformu *PC Magazine* međutim smatra da su oba procesora na ovom polju poredive i da je glavni krivac za katastrofalan rezultat loš kompajler za 32-bitni kod Pokazalo se takođe da je Pentium mnogo brži u 32 bitnoj komunikaciji sa diskom a da PowerMac ima mnogo bržu grafiku Razlog za slab rezultat je pronađen u *System*-u *7.x* koji za razliku od *Windows NT*-a na koristi slobodnu radnu memoriju za keširanje diska

Brzina nove Apple mašina bila je još manje ubedljive u svetu stvarnih aplikacija. Izvršavajući *PhotoStyler* PowerMac je bio izrazito brži ali (iznenađenje) najbrži PowerMac je bio sporiji od Pentiuma u izvršavanju namenske verzije programa *PageMaker* Na testovima Intel i *Windows* emulacije PowerMac se takođe nije proslavio razvijajući samo 13 odnosno 30 odsto snage Pentiuma na 66 MHz Apple međutim spremno izjavljuje da daje sve od sebe da i ovde popravi efikasnost svojih mašina.

TEHNOLOGIJE: LICEM U LICE

Prvim odmeravanjem snaga, međutim, nisu zadovoljni ni Intel ni IBM (koji praktično jedini proizvodi PowerPC čipove) Intel zato što u poređenjima ne prolazi kako bi želeo a IBM Apple i Motorola zato što osećaju da mogu i bolje Intel je stoga požurio da izbaci novu generaciju Pentium procesora u novoj tehnologiji sa debljinom slojeva od 0 6 mikrometara napajanjem od 3 3V i unutrašnjim taktom koji je za 1 5 puta veći od spoljašnjeg (90/60 i 100/66 MHz) Površina čipa je smanjena na polovinu a potrošnja energije na trećinu Suprotna strana je reagovala vešću da su proizvedeni prvi uzorci „maloitraznog" modela 603 koji je dovoljno štedljiv za ugradnju u *notebook* računare kao i sledećeg člana PowerPC porodice modela 604

PowerPC 604 će biti izbačen na tržište već tokom narednih meseci Procesor radi na 100 MHz Ima šire komunikacione linije (*pipline*) viši stepen paralelizma, unapređenu tehniku predviđanja grananja i dovoljnu rezervu snage smatraju analitičari da tuče svaku Pentium mašinu u narednih 18 meseci Od Pentiuma na 100 MHz je brži 60 odsto i trebalo bi prema prvim analizama da izvršava x86 softver u emulacionom režimu istom brzinom kao i Pentium na 66 MHz I na samo to Sledeće godine se očekuje verzija od 150 MHz u tehnologiji od 0 5 a možda i od 200 MHz u tehnologiji od 0 35 mikrometara Očekuje se da će verzija od 100 MHz razviti 160 verzija na 150 MHz 230 a verzija na 200 MHz 300 SPECint92 jedinica pri čemu bi poslednji rezultat trebalo da bude poređiv sa Pentium procesorom sledeće generacije Sadašnji Pentium na ovim testovima pokazuje znatno slabije rezultate – verzija na 100 MHz ima 100 a verzija na 150 ca imati 150 SPECint92 jedinica Zapravo prve analize govore da Pentium nema gotovo nikakvih šansi u poređenju sa PowerPC 604 na 100 i DEC Alpha čipom 21064A na 275 MHz. Tako će u novoj iteraciji razlike između PowerPC i Pentium procesora postati još očiglednije

Kako se međutim porede same arhitekture ova dva najdirektnije suprodstavljena čipa?

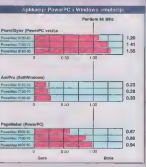
Model 601 je prvenac u PowerPC familiji i nastao je prostim spajanjem IBM RSC procesora Motorolinog čipa 88110 za kontrolu magistrale Izvršava do tri instrukcije u jednom ciklusu (ako se nalaze u različitim jedinicama – za integerske instrukcije za aritmetiku u pokretnom zarezu i za grananje u programu) skok i operacija sa racionalnim brojevima se može izvršiti sa bilo koje prve četiri pozicije u redu za instrukcije, a keš kapaciteta 32 K je zajednički i za insturkcije i za podatke Pentium ima dva linije za integerske instrukcije ali u jednom ciklusu izvršava samo posebne pareve instrukcija. Predviđanje skokova je zahvaljujući posebnom baferu u kojima se čuvaju rezultati za 256 instrukcija znatno efikasnije a razdvojeni keš za instrukcije i podatke ima i svojih prednosti ali i – mana Instrukcije i podaci nisu pomešani kao kod jedinstvenog keša ali se zato može narušiti ravnoteže između onoga u kome se čuvaju instrukcije i onoga u kome se čuvaju podaci Ključna prednost PowerPC familije je međutim što za približno istu snagu koristi dvostruko jeftiniju tehnologiju

PREOKRET: KAD IBM OKRENE LEĐA

Snažna softverska (i medijska) podrška konkurentskog proizvoda nije najstrašnije što je moglo da zadesi Intel ovih dana Leđa mu jedan za drugim okreću i proizvođači hardvera opredeljujući se u 486 klasi za AMD i Cyrix a na Pentium sceni za PowerPC i Alpha procesore Najpouzdaniji partner i najveći kupac Intelovih procesora IBM povukao je u poslednje vreme nekoliko veoma neprijatnih poteza Prihvatio je da proizvodi rivalski Cyrix 486 procesor odustao od svojih prava da proizvodi Pentium i čipove potonje generacije i počeo više nego upadljivo da merka PowerPC 604 kao najozbiljnijeg kandidata za svoju PC liniju Drugi partner po veličini i značaju Compaq odlučio je da sa svojih mašina skine logo „Intel Inside" i pređe na AMD čipove Digital Equipment uskoro počinje s proizvodnjom AMD procesora i naravno s njihovom ugradnjom u svoje PC mašine Intel je godinama bio jedna od najprofitabilnijih kompanija u svetu sa veoma zdravom marginom dobiti Sada je vrednost Intelovih deonica počela da pada (trenutno sa 5.20 na 5 20 dolara) a margina dobiti da se sužava. Ako želi da se suprodstavi procesoru PowerPC i drugim RISC čipovima s jedne i agresivnoj konkurenciji na 486 polju s druge strane Intel će smatraju finansijski analitičari morati znatno da – pojeftini

"Nama oduvek prete RISC i oduvek nam prete imitacije" izjavio je nedavno Intelov predstavnik „Naš odgovor je da na tržištu budemo brži sa boljim performansama." Uprkos svojim (samo)ohrabrujućim izjavama da nije zaplašen Intel je odista veoma zabrinut očiglednim uspehom PowerPC procesora Ovogodišnji CeBIT umalo nije završen skandalom kada je uzrujani predstavnik Intela pokušao da spreči uporedni javni test Pentium i PowerMac mašina. IBM je na svom štandu poredio PowerMac i Pentium PC sa istim *Windows 3 1* softverom dok je Motorola uporedivala obe mašine pod *Windows NT* operativnim sistemom ali je Intel uz pomoć svojih advokata pokvario ovu priredbu uz obrazloženje da takve vrste poređenja nije dozvoljena u Nemačkoj Osnovna svrha poređenja je bila da pokaže da je Pentium najzad dobio alternativu i Intelova žučna reakcija je možda najbolji znak da se to i dogodilo Ova smirujuća izjava Motorole koja se očigledno pravi nevešta nikoga ne može da zavara pogotovu ne uzrujanog Intela. Svi znaju da se proizvođači PowerPC procesora neće zadovoljiti samo time da pokažu kako sto sve može i drukčije Sasvim je sigurno da će oni uzeti od tržišta sve što budu mogli da dobiju i da zapravo ciljaju na sve

Svoju nadu u kontinuirani poslovni uspeh Intel polaže kako u nove Pentium modele u 0 6 mikrometarskoj tehnologiji na 90, 100 i 150 MHz (očekuje se da će ove godine proizvesti između 6 i 8 miliona Pentium procesora) tako i u ogromnu bazu aplikacija za x86 platforme Brzina emulacije aplikacija za Intel platforme na PowerMac mašinama još se Intel frustrira svakog korisnika DOS i *Windows* programa kome su potrebne iole bolje performanse Oslanjajući se na rezultate svog sopstvenog istraživanja Intel tvrdi da se u 64 odsto PC mašina u kućnoj primeni nalazi 486 procesor i to uglavnom visoke klase kao i da se u 85 odsto svih računara koji su ikada prodati nalazi Intelov procesor Zaključak je po Intelu jasan oni koji ne mogu bez DOS-a i *Windows*-a neće moći ni bez – Intela



Testiranje je izvršeno programom *MacBench* koji je pisan u visokoprenosivom C-u i koji se može pokrenuti praktično na svim platformama (*Windows Windows NT*, *OS/2 System 7*) U Pentium računaru se nalazila grafička kartica sa čipom WD90C33, a u PowerMac mašinama, izuzev najnižeg modela kod kojeg je korišćena grafika na matičnoj ploči, AV kartice Loši rezultati u radu sa racionalnim brojevima se pripisuju ograničenjima Apple C kompajlera, a podbačaj u radu sa diskom ograničenjima operativnog sistema *System 7*, koji još uvek dolazi u 68040 kodu i koji, za razliku od *Windows NT*-a ne koristi slobodnu memoriju u sistemu za keširanje diska.

SMENA ILI ALTERNATIVA?

Istorija računarstva je puna velikih projekata koji su padali u zaborav gotovo odmah posle velikih pozdravnih govora Stvari ovoga puta izgledaju previše ozbiljno Apple je lansirao prvi personalni PowerPC računar kojim planira da se milion komada, u narednih godinu dana u potpunosti preplavi tržište Motorola i IBM su napravili specifikaciju platforme i uspeli da za nju zainteresuju proizvođače sa Tajvana kao i proizvođače kompjuterskih komponenata Softverska industrije sa oduševljenjem prihvata novu platformu i ističe njene prednosti u odnosu na Intel procesore Sudeći prema do sada objavljenim testovima PowerPC u prvoj iteraciji ima veću sirovu snagu nego Intel sa svojim Pentiumom, u poslednjoj

Šta je pored ovoga još potrebno za uspeh na tržištu? Možda je sve već tu Možda se upravo ovih dana priprema velika istorijska smena generacija u kojoj će već pomalo potrošena CISC tehnologija, sa krajnje oskudnim rezervama snage ustupiti mesto potpuno svežem RISC-u čije rezerve koje će se polako oslobađati sa novim iteracijama u ovom času izgledaju neiscrpne Za korisnike personalnih računara, međutim ova smena generacija će po svemu sudeći, biti potpuno bezbolna Jednog dana će možda doći i zameniti nam procesor u računaru a da to nećemo ni osetiti Za sada međutim niko nema smelosti da priča o smeni Svi govore samo o – alternativi



MADE IN AMERICA

Dobiti posao u Americi je prava mala umetnost. To se ne uči preko noći, baš kao što ni vaš engleski neće postati besprekoran preko noći. Proći će meseci, možda i godina dok ne nađete posao u struci. A dotle... Dotle ćete morati da radite sva i svašta. Kad sam mogao ja, možete i vi.

Vlada Kostić

Kako stići u Ameriku? Teško. Postoji nekoliko načina.

1. Ako ste jako srećni, probajte da tražite vizu u Američkoj ambasadi. Može da upali, mada je vrlo malo verovatno. Postoje razne vize, u principu je najlakše dobiti turističku. Potrebno je da neko garantuje za vas – ne tražite to od mene, jer nisam američki državljanin. Obično pomaže ako ste student ili zaposleni.
2. Meksička granica. To nisam probao i ne znam kako funkcioniše, osim da je izuzetno rizično. Bela koža je na ceni.
3. Ukrcati se u avion i onda na licu mesta kukati kako ne možete nazad zbog situacije u YU (građanskog) rata itd. Jednostavno tražite azil. Pre godinu-dve dana to je vrlo dobro funkcionisalo. Da bi vas izbacili iz zemlje, mora sudija da done se odluku – ne mogu carinski radnici. Na suđenje se čeka bar godinu dana. Dotle bi trebalo da vas stave u zatvor. Ali pošto su zatvori prepuni, šanse su 99,9% da vas šutnu na ulicu sa nalogom da se pojavite pred sudom jednog lepog dana. A dotle... Ovaj način je odlično funkcionisao do pre nekog vremena, naročito u Njujorku, ali sam čuo da su jako zategli propise. Ako krenete tim putem, radite to na svoj sopstveni rizik. Naravno, dodatni problem je ukrcati se u avion bez viza. Obično avionske kompanije ne vole da imaju problema sa Američkom imigracijom i proveravaju da li imate vizu ili ne pre nego što vas puste u avion.
4. Preko Kanade. To ne znam kako funkcioniše, a svakako je potrebna viza za Kanadu, što je veoma teško dobiti.
5. Nekada se lako ulazilo u Ameriku preko studentske vize. Jednostavno se upišete na neki od ovdašnjih univerziteta. Aha, samo postoje dva sitna malecka problema.

   Prvo, američki univerziteti koštaju ludački puno. Ako je verovati Tamari (koja vas sve pozdravlja), potrebno je oko $10 000 da joj se pridružite kao redovan student.

   I drugo, studentske vize ne daju više nikome iz YU – zbog sankcija. Ako jednom ukinu sankcije... Ko zna?

JEDNOM U AMERICI...

New York! Manhattan! Žuti taksiji jure ulicama. Oblakoderi. Crnci prodaju viršle i Coca-Colu ispred skupocenih izloga. More ljudi žuri na posao. Kuda sada?

Pomaže ako imate kod koga da odete. Takođe jako pomaže da imate bar elementarno znanje engleskog jezika. Bez toga ne polazite!

Ako nemate nikoga ovde, treba vam najmanje $5000 da krenete. Bez toga spavaćete na ulici. Vrlo neprijatno.

Smeštaj Najjeftinije je iznajmiti sobu. Zato kupite novine (25 centi) i pogledajte oglase. Telefon je neophodan! Ne samo da se javite na oglas, nego da se vama jave! Obično će vas sačekati telefonska sekretarica sa porukom: „Žao nam je što nema nikoga da vam odgovori. Ako tražite sobu, ostavite svoj broj telefona i pozvaćemo vas." Zato vam treba telefon, a ne govornica. Dakle, prvo uzmete sobu u hotelu (sa telefonom!) – to košta oko $50 dolara za noć – i onda tražite sobu. Soba košta oko $300–450 mesečno, plus depozit od oko $200.

Stan Obično vam niko neće iznajmiti stan ako nemate pristojan posao i kreditnu istoriju. Izuzetak su loši stanovi u crnačkim krajevima koje niko neće. Stan košta od $500 naviše (mesečno). Isplati se deliti stan sa nekim. Recimo jednosoban stan košta $750. Trosoban u istoj zgradi samo $250 više. Korak dalje je kupovina kuće (na kredit, naravno). To ne znam kako se radi – još nisam stigao dotle.

Hrana Ako ne znate da kuvate, hrana će vas koštati od $5 do $15 dnevno. Isplati se kupiti gotovu smrznutu hranu koja se samo stavi u rernu ili *microwave* i bude gotova za par minuta. Takođe je jako praktično kupovati plastične sudove – ne mislite valjda da ja perem sudova?

Kola Bez kola je nemoguće živeti u Americi. Izuzetak je jedino centar Njujorka, ali vas onde stan košta ludački puno. Javni prevoz jednostavno ne postoji, a razdaljine su divljačke. Ja, recimo, radim 40 kilometra od mesta gde živim i to je za ovdašnje uslove još i blizu. Za početak neka polovna kanta može da posluži ($1000 pa naviše). Ne isplati se kupiti neki totalni kiš ($500) jer se kola moraju registrovati i neće proći tehnički pregled. Gledajte da vas prodavac polovnih kola ne nasanka! Registracija i tablice koštaju oko $200.

Ne vredi da iznajmite kola. Prvo, to puno košta. Drugo, niko vam neće iznajmiti kola ako ne platite kreditnom kartom. Kreditnu kartu nećete dobiti bez dobre kreditne istorije, a to nećete steći preko noći.

Vozački ispit To je bar lako! Ovde čak i stare bakice voze kola. Vozački ispit se polaže za dva sata i odmah vam na licu mesta daju vozačku dozvolu. Potrebno je da imate pasoš i *social security card*. Ako ne možete da položite vožnju iz prvog pokušaja, onda ste stvarno težak debil. Sa internacionalnom dozvolom možete da vozite 6 meseci – ne duže.

Social Security Card To je potrebno da se dobije vozačka dozvola, a i sve ostalo (banke vas, recimo, vode u svojim kompjuterima po „social security" broju, a ne po imenu i prezimenu). *Social security card* se dobija tako što odete u *Social Security Administration* i tražite *social security card* radi vozačke dozvole. Ništa lakše.

Vozačka dozvola To je ovde univerzalan dokument – kao lična karta kod nas. Kada jednom dobijete vozačku dozvolu, možete slobodno svoj jugoslovenski pasoš da bacite u đubre – više vam neće trebati.

Osiguranje Nećete registrovati kola bez osiguranja. Osiguranje će vas koštati oko $1000 godišnje, a plaća se na mesečne rate. Pazite, svaka kazna za brzu vožnju podiže osiguranje u nebesa.

Vožnja O tome neću da komentarišem, osim da voze loše i jako sporo. Pazite na brzinu! Američka policija strogo kontroliše brzinu i kazne idu od 50 dolara do 1500, do zatvora. 55 milja (90 Km/h) je dozvoljena brzina na autoputevima. Kazna za brzu vožnju, osim što košta, osim što se smatra za krivično delo – ne prekršaj. Krivična dela vam neće baš pomoći da sredite papira za boravak. Pazite, glupi Ameri ne poštuju pravilo da je leva traka samo za pretacanje. Takođe menjaju trake bez žmigavca i ne gledaju u ogledala. Voziti brzo u takvim uslovima je izuzetno opasno! Pijane vozače šalju pravo u zatvor – pazite da vas to ne snađe! Benzin nije skup – oko $1.20 galon (3.8 litra). Putevi i ulice su odlično obeleženi.

Telefon Šta god da vam treba, samo prilisnete nulu i javi se „operator". Onda pitate šta i kako. Kao što sam već rekao, svoj telefon morate da imate. Ne samo da nađete sobu, nego i za posao i sve ostalo. Ako iznajmite sobu (ili stan), uvođenje telefona košta oko $60 (plus depozit od još jedno $50). Telefon (sa sekretaricom) košta oko $100. Telefonski imenik je takođe jedna od neophodnih stvari za opstanak.

Kriminal Kriminal je ovde stravičan! Ako se zateknete gde ne treba, a još posumnjaju da ste stranac... Obično kad stignete u neki grad prvo pitate gde ne valja zalutati. Vrlo je zanimljivo kako je kriminal strašan u nekim oblastima, a vrlo mali samo malo dalje. Recimo u Vašingtonu će vam obiti kola da ukradu 25 centi koji su vam ispali iz džepa i ostali na sedištu. Samo 15 kilometra dalje (u predgrađu) meni je walkman ležao na zadnjem sedištu (nezaključanih) kola tri meseca i niko ga nije ukrao.

Uobičajena je praksa za Amerikance da uvek nose oko 20 dolara u džepu – ako ih neko napadne. Ako vas napadnu, a nemate ništa da im date, mogu lako da vas izbodu nožem ili upucaju.

Ovakav kriminal je posledica blagog pravnog sistema. Verovatno najblažeg na svetu. Posledica? Ljudi voze po gradu sa zaključanim vratima. Hodaju ulicom sa strahom u očima. Žene ne idu nikuda same. „Odbrambena" industrija cveta – sprej sa suzavcem, razne pištaljke, alarmni sistemi, oružani čuvari itd.

Pare Cela Amerika se vrti oko para. Ne postoji stvar koja se ne može dobiti za pare. Takođe ne postoji stvar koja ne košta. **Neće vam dati ni da dišete besplatno!**

Seks Zatucaniju naciju na svetu nisam video. Nije da su konzervativni, nego to nije normalno! Seksualna revolucija? To mora da je dobra šala! Amerikanke (plavuše) koje se dobro ***? O tome mi sanjamo u Evropi! To ovde treba svećom tražiti. Doduše, otkrio sam u zadnje vreme da voziti Porsche-a 944 pomaže kod plavuša.

Religija Isti slučaj kao sa seksom. Žive ljudi u 17. veku. Šta se može?

Nasilje Taman posla da na američkoj televiziji vidite neku malo slobodniju seksualnu scenu. Ali nasilje, pucanje, ubijanje, pljačkanje itd. – za to nema problema. Čak i u emisijama za decu.

TV Mislite da će vam dati da gledate televiziju besplatno? Naravno da neće! Besplatno ćete gledati samo suve reklame. Prava televizija je kablovska televizija i to, naravno, košta – oko $50 mesečno.

Kultura Zeleni dolar je sve što od kultura imaju.

Advokati U Americi ima više advokata nego u svim ostalim zemljama sveta zajedno. Stvarno. Ta gamad će vam popiti krv... nećete to izbeći. Budite upozoreni.

Doktori Eh, to košta! Više nego sami advokati. Razboleti se u Americi je sport bogatih. Kratak pregled kod doktora (15 minuta) košta $100. Kutijica antibiotika $70. Boravak u bolnici zna da košta $100 000 i više. Ako vas, ne daj bože, zaboli zub, ni jedan zubar vas neće pogledati dok mu ne mahnete gomilom stotki. Imati dobar posao pomaže – firma plaća zdravstveno osiguranje, ali i dalje morate da pokrijete deo (astronomskih) troškova iz svog džepa.

Kreditna istorija Loša kreditna istorija je drugi najveći maler koji može da zadesi Amerikanca. Prvi najveći maler je poreski inspektor IRS-a na vratima. O IRS-u kasnije. Gotovo sve se ovde kupuje na kredit. Sa lošom kreditnom istorijom nećete moći ništa da kupite. Za one koji dolaze u Ameriku, problem nije loša kreditna istorija, nego nedostatak bilo kakve kreditne istorije. To nije mnogo strašno, ali znači da treba steći kreditnu istoriju. Sa kreditnom istorijom će doći kreditne kartice i sve pogodnosti koje to donosi. Potrebno je par godina da se stekne kreditna istorija. Kreditna istorija se uništava tako što se propusti plaćanje jedne rate na vreme, i još gore, ne plati se ceo kredit ili neki račun ili bilo šta drugo. Sa parama nema zezanja u Americi.

Pošto je bar jedna kreditna kartica (Visa) apsolutno neophodna u Americi, i pošto je nećete dobiti bez dobre kreditne istorije, savetujem sledeće: uzmete kreditnu karticu sa limitiranim kreditom od $500 koja se dobije nakon što se deponuje $500 dolara. Jeste da nema smisla deponovati $500 da bi se dobio kredit od $500 na koji se onda plaća kamata, ali to je najbrži način da se započne kreditna istorija.

Čekovni račun Ima stvari koje se jednostavno ne plaćaju gotovinom, nego isključivo čekom. Zato vam treba čekovni račun. Srećom, to se dobija za 10 minuta u svakoj banci. Takođe treba vam ATM kartica da podižete novac sa automata kada vam zatreba.

Američko društvo Na svakom koraku neko pokušava da vam otme novac legalno ili ilegalno, otvoreno ili prefingano, pošteno ili na prevaru.

Politika Amerikom vlada takva klika beskrupuloznih i za novac krvožednih političara da vam se od toga povraća. Samo 10% (naivnih) Amerikanaca ima poverenja u vladu.

Neprijatelji Vaša dva najveća neprijatelja u Americi su INS (*Immigration and Naturalization Service*) koji bi da vas deportuje i IRS (*Internal Revenue Service*) koji bi da vam otme sve para dok ste još tu.

KAKO SREDITI PAPIRE?

Jako teško. Nećete ama baš ništa uspeti bez advokata, a advokati će vas odrati živog (teške hiljade i hiljade dolara). Siguran put je ženidba, ali nećete naći Amerikanku da se oženi besplatno, a to i kasnije ima da ima ružne (finansijske) posledice kada dođe do razvoda. Ženidbom nećete izbeći advokate – trebaće vam i dalje da sredite papire.

Papiri i sve ostalo oko boravka u Americi se rešava u INS-u (*Immigration and Naturalization Service*).

Zelena karta Zaboravite na to za početak. To se ne dobija ni brzo ni besplatno. Za početak vam treba samo radna dozvola. Zelena karta je za one koji na znaju, potvrda da ste stalni stanovnik Sjedinjenih Američkih Država i to je poslednji korak pred puno državljanstvo.

Radna dozvola Postoje razni načini da se dobije. Možete, recimo, da tražite B1 ili B2 vizu što dozvoljava da radite. To možete da tražite u Američkoj ambasadi (ne preporučujem) ili tek kada stignete ovde (što je bolje). Treba neko da vam garantuje posao (a to nije lako!) i treba da imate završen fakultet. Takođe pomaže ako znate da radite posao za koji ovde ima malo stručnjaka. (Ne, to nisu kompjuterski programeri, ima koliko hoćete.) Opet, budite spremni da platite na hiljade dolara advokatima!

Takođe radne dozvole se dobija ako prime vaš zahtev za (politički) azil. Znate ono ... rat i tako to. Ne moraju da vam odobre azil – taj postupak traje godinama, ali dok god razmatraju zahtev vi imate radnu dozvolu i možete da radite. Šanse da prime zahtev za azil su ravne nuli. Ja sam probao ... i uspeo! Loša strana je što jednog dana mogu da donesu odluku da se zahtev odbija (što je sto puta verovatnije nego da se prihvate). A onda...

Čuvajte se prevaranata Znam ljude koji su pukli silne pare raznim prevarantima koji su **garantovali** zelenu kartu, ili radnu dozvolu, ili azil.

A ŠTA BEZ PAPIRA?

Jednom kada uđete u Ameriku niko vas neće izbaciti van – pod uslovom da ne pravite probleme sa policijom. Do sada je to uglavnom bilo tako, mada sam čuo i za Jugoslovene koje je INS deportovao. Ako uspete da nađete posao na crno, možete sasvim lepo da živite. Na crno se može naći samo jako loš i malo plaćen posao. Naravno, bilo bi vam mnogo bolje sa papirima, ali šta se može?

Doskora je bilo lako naći posao bez papira – poslodavca je interesovalo da li umete da radite posao ili ne. Pitanje vašeg imigracionog statusa je bio problem INS-a, a ne poslodavca. Sada to više nije tako. Donesen je zakon da poslodavac odgovara ako zaposli ilegalnog imigranta (ovi to zovu *illegal alien*). Zato je sada bez papira moguće naći samo loš posao koji niko neće.

Čuo sam priče da ima država gde ne paze mnogo na papire, ali to nisam provero. U tri države gde sam probao (Maryland, Virginia i Washington DC) nemoguće je za programere da se zaposli bez ispravnih papira.

Takođe možete da falsifikujete papire. Za legalan posao je potrebno pokazati vozačku dozvolu i *social security card*. Na vašoj *social security* karti će pisati „Not Valid for Employment" jer niste američki državljanin. Ako to falsifikujete, a engleski dovoljno dobro govorite da se prošvercujete kao Amerikanac, rešili ste problem – bar možete da nađete dobar posao.

Postoji još sto drugih načina da se papiri falsifikuju, ali lično ne savetujem igranje sa falsifikovanim papirima. Ako nemate papira i pravite se manji od makovog zrna, niko vam neće ništa. Može INS da vam preti svim i svačim, ali će teško preduzeti nešto konkretno da vas deportuje. Međutim, ako vas uhvate sa falsifikovanim papirima...

POREZ

Izbeći porez je u Americi veći greh nego ubiti čoveka, a redovno plaćanje poreza neophodan preduslov da se sredi papir za stalan boravak. Porez odbija poslodavac od vaše plate i uplaćuje IRS-u. To boli! U mom slučaju odbijaju (otmu!) preko $1 000 svakog meseca.

Svakog aprila ste dužni da popunite obrasce za porez (koliko ste zaradili, koliko vam je poslodavac odbio od plate itd.) i to pošaljete IRS-u. Svako igranje sa IRS-om se skupo plaća zatvorom i to na savetujem nikome. U principu, odbijaju malo više od plate nego što bi trebalo. Zato vam IRS, nakon što u aprilu prijavite porez, pošalje ček na račun viška koji su uzeli (oteli).

Međutim, postoji i mogućnost da vi dugujete novac IRS-u. To se često dešava kada radite za neku firmu kao spoljni konsultant. U tom slučaju firma ne odbija porez od vaše plate – to morate sami da uradite i novac uplatite IRS-u. Prošle godine sam zaradio $15000 radeći kao spoljni konsultant. A onda me je u aprilu sačekao IRS – dugujem im $4000.

A šta ako radite na crno? Ima ljudi koji kažu da se u tom slučaju treba praviti lud i ne plaćati porez. Ima i onih koji tvrde da IRS baš briga da li radite legalno ili na crno – hoće svoj porez i tačka. IRS nema nikakve veze sa INS-om i njih ne interesuje vaš imigracioni status.

Interesantan je i slučaj ako nemate papire, a poslodavca ste slagali da ih imate. Naravno da u tom slučaju poslodavac odbija porez od vaše plate na račun IRS-a. U tom slučaju savetujem da se prijavi porez – čisto da dobijete neku kintu nazad od IRS-a.

Porez se plaća IRS-u (*federal tax*) i lokalnoj državi (*state tax*), koja nije ništa manje pohlepna od samog IRS-a. Ženidbom nećete sebi smanjiti porez. To je nekada palilo, sada više ne. Kupiti kuću je sada bolje rešenje.

Ako neko zna način kako da **legalno** izbegnem porez, neka mi javi. Jako me zanima.

KAKO NAĆI POSAO?

Jako, jako teško. Ja sam mislio da će me kao izuzetnog stručnjaka za kompjutere svi dočekati obe ručke. Malo sutra!

Prvo i prvo, potreban je rezime (*resume*). Ne pokušavajte da napišete sami svoj rezime – nećete ništa uraditi. Umesto toga uzmete novine, otvorite sekciju sa oglasima za posao i nađete firmu koja se bavi pisanjem rezimea. Platite im $200, kažete im ko ste i šta ste i šta umete da radite i dobijete lepo napisan profesionalni rezime, obično na dve strane (vlada mišljenje da rezime treba da bude dve strane). Ja sam otkrio da u mom slučaju najbolje pali rezime od 5 strana.

Evo šta treba da bude u rezimeu:

Ime, prezime, adresa i telefon, šta umete da radite, gde ste i šta radili, koju diplomu imate, lista referenci.

U rezimeu ne treba navoditi koliko godina imate, gde ste živeli, kako vam se zove mama.

Rezime treba da je sažet, pregledan, jednostavan i atraktivan. Nakićen i prenatrpan rezime odštampan sitnim slovima ne valja. Rezime je od suštinskog značaja da se dobije posao! Obično stvari idu ovako: Firma koja hoće da zaposli stručnjaka za kompjutere da oglas u novinama. U roku od 2 nedelje dobiju na stotine, ponekad hiljade rezimea. Od toga izaberu 10 najzanimljivijih i pozovu te ljude na razgovor. Od tih 10 jedan dobije posao.

Dakle, rezime mora da je **apsolutno besprekoran**, inače vas neće pozvati ni na razgovor, još manje vam dati posao. Postoje dva strategije da se nađe posao: 1. Koristite svoje kontakte i poznanstva. To dobro pali ako ste priznat stručnjak i ako poznajete puno ljudi u struci. Međutim, kada stignete u Ameriku ne znate nikoga, niti iko zna za vas. 2. Preko oglasa u novinama. Obično u nedeljnim novinama ima sekcija sa stotinama oglasa za posao.

U Americi stručnjaka za kompjutere ima koliko hoćete. Zato niko ovde ne zapošljava univerzalnog stručnjaka već specijalizovanog stručnjaka. Logika je sledeća: Treba nam stručnjak za C++. Dakle, zaposlićemo stručnjaka za C++. Ako nam sutra treba stručnjak za D--, šutnućemo našeg stručnjaka za C++ i zaposliti stručnjaka za D--.

To morate da imate na umu i kada šaljete rezime i kada idete na razgovor. Ako im treba C++ stručnjak, pričaće vas šta znate o C++ i koliko ste godina radili na C++. To što vi znate još 10 drugih programskih jezika njih uopšte ne zanima. Zato savetujem da imate ne jedan rezime, nego nekoliko rezimea. U jednom se predstavite kao C++ stručnjak, u drugom kao FOXPRO stručnjak, u trećem kao DB2 stručnjak itd. Onda koristite rezime koji odgovara oglasu.

Pod uslovom da je rezime besprekoran, od 100 koje pošaljete dobićete samo jedan poziv na razgovor, i to obično posle par nedelja. Dakle, treba se naoružati strpljenjem i slati na stotine rezimea. Na razgovor se ide u odelu i sa kravatom. Tako se, uostalom, ide na posao svaki dan. Farmerke na dolaze u obzir. Dobro poznavanje engleskog jezika je apsolutno neophodno. Ne morate da imate savršen akcenat (ja ga sigurno nemam), ali morate odlično da govorite i pišete.

Sam razgovor treba voditi vrlo mudro. Prvo vidite šta im treba, a onda se predstavite kao stručnjak baš za tu oblast. Možete slobodno da lažete – dok god ste u stanju da uradite posao. Inače – šutnuće vas sa posla po kratkom postupku čim otkriju da ne umete da radite.

Ne, ne vredi da se predstavite kao genije, čak i ako to jeste. Recimo, firma u kojoj sada radim je tražila stručnjaka za „client-server architecture". Ja naravno nisam znao ništa o „client-server architecture", ali sam se napravio lud, napisao rezime u kojem se predstavljam baš kao stručnjak za to, otišao na razgovor, ponovo se predstavio kao stručnjak koji je to radio godinama, i ... dobio posao! Onda sam po najkraćem postupku naučio sve što ima da se zna o „client-server architecture" i sada sam stvarno stručnjak za tu oblast. Međutim, da sam im rekao „ja ne znam ništa o tome, ali ja sam genije, verujte mi na reč i ja ću to da naučim do sutra", malo sutra bi mi dali posao.

Dobar fakultet pomaže. Međutim, za Amerikance dobar fakultet je MIT ili Berkeley, a ne beogradski ETF. Ako ste stvarno izuzetan stručnjak, onda vas neće baš puno ispitivati za školu. Kada im kažem šta sve umem da radim i šta sam sve radio i koliko dugo, niko me ne pita za fakultet. Istine radi, nisam završio ni srednju školu (stigao sam do trećeg razreda).

Dodatni problem su reference. Naravno da ne pomaže ako im kažete „Ja sam radio za sve ove silne firme u Jugoslaviji". Trebaju vam američke reference, a to se ne stiče preko noći. Za početak treba naći: 1. Firmu kojoj toliko odgovarate da će preći preko vašeg nedostatka referenci. To je vrlo malo verovatno. 2. Firmu koja će preći preko vašeg nedostatka referenci ali vam za uzvrat daju nikakvu platu. Za prvi posao koji sam našao u struci dali su mi platu od $16 na sat što je za taj posao bilo krajnje mizerno. To zato što mi engleski nije bio besprekoran i zato što nisam imao reference i zato što nisam znao sve štosove kako napisati rezime i kako se ponašati na razgovoru. Danas zarađujem dve puta više.

OBEĆANA ZEMLJA

U Americi se radi od 8 ujutru do 5 uveče – u principu. To često ume da bude od 8 ujutru do 8 uveče. Ako niste spremni da stvarno radite, bolje ne polazite ovamo!

Dobar posao je prava mala umetnost. To se ne uči preko noći. Takođe ni engleski vam neće postati besprekoran preko noći. Zato će proći meseci, možda i godine dok ne nađete posao u struci. Dotle... Dotle ćete morati da radite sve i svašta, uključujući i fizičke poslove. Kada sam mogao ja, možete i vi.

Obećana zemlja? Malo sutra! Ova zemlja je puna problema i tek će da pukne. Društvo, ekonomija, kriminal, rasno pitanje, državni deficit itd. Sve ce to da eksplodira u narednih desetak godina. Međutim, ne znam ni jedno mesto na planeti Zemlji gde bi programeru iz Jugoslavije bilo bolje nego u Americi. Bar za sada.

Složite kocke, povežite Vaše računare u *NET*®



NET®

Computer Engineering

Braće Ribnikar 52, 21000 Novi Sad, tel: 021/616-977, fax: 021/615-944

DESIGN BILUX

SVE VEĆI, SVE JEFTINIJI

Ništa u poslednje vreme nije tako brzo rasio po kapacitetu i tako brzo padalo po relativnoj ceni kao tvrdi diskovi. Nikada ranije megabajt nije koštao toliko malo. Pokazujemo to na primeru osam diskova, a svi se mogu naći na domaćem tržištu.

Dejan V. Veselinović

Svi smo svedoci velikog bujanja softvera koji zahteva sve veći prostor na disku. Čak i svaka nova verzija tekst-procesora danas zahteva više puta pomnožen prostor koji je tražila prethodna. Ako se opredelite za *MS Word for Windows 6.0* trebaće vam nekih 18 MB za *Windows* i daljih 10+ MB za sam program i nekoliko fontova, a ako poželite još fontova trebaće vam još nekoliko desetina megabajta. Čak i ako se opredelite za manje zahtevan *WordPerfect 6.0 for DOS* opet računajte da ćete za njega odvojiti nekih 14-18 MB (u poređenju sa 3,5-4.3 MB za verziju 5.1).

Sve to, uz opšti trend prelaska na grafička okruženja, nameće sve veće i veće zahteve za prostorom. Toj činjenici se prilagođava i industrija diskova, koja ubrzano izbacuje iz proizvodnje manje modele, a dojučerašnji modeli snova (delimično i zbog dojučerašnjih astronomskih cena) danas postaju standard osnovna ulaznica za bal bez dodataka. Prvo su stradali diskovi ispod 100 MB, a sada se povlače i diskovi ispod 200 MB.

To nas je navelo da se malo pozabavimo novim modelima diskova koji se kod nas nude – to i činjenica da se podože nismo bavili diskovima kao takvima. Za ovaj put, probali smo osam diskova iz dva izvora i od četiri proizvođača.

Maxtor 7345AT

Za razliku od zlosrećnog modela 7060A koji je dosta nagrizao renome ove poznate kuće (kao svojevremeno i Seagate model 237) ovaj disk spada u novu generaciju ovog proizvođača koji je očigledno odlučio da napravi povratak na velika vrata. Disk ima 790 cilindara, 15 glava i 57 sektora po tragu.

Proizvod deluje veoma solidno, pločica sa elektronikom je besprekorna. Jedino što pomalo čudi je relativno mali bafer od svega 64 KB. Inače ovaj disk ima dvostrano (*dual port*) višesektorko baferovanje i čitanje podataka. Veličina bafera je 128 sektora, podržava DMA, podržava pisanje u režimu dvostruke reči. Brzina prebacivanja glave sa traga na trag je kraća od 15 mikrosekundi (što ni jednom drugom disku u ovoj grupi nije pošlo za rukom). No, zato je ovo jedini model na kome nije navedena potrošnja struje, sam bog zna zašto, bilo na njemu samome, ili u propratnoj literaturi.

Vizuelno se razlikuje od manjeg brata 7213A po poliranom poklopcu od anodizovanog aluminijuma, umesto klasičnog crnog poklopca poznatog od ranije.

WD Cavier 1210 i 1270

Iako pripadaju istom proizvođaču, ova dva diska su ipak dva modela, a ne dva podverzije iste platforma. To se vidi po nizu konstrukcionih detalja, od kućišta do elektronike. Model 1210 ima 989 cilindara, 12 glava i 35 sektora po tragu, a model 1270 ima 917 cilindara, 12 glava i 48 sektora po tragu.

Diskovi podržavaju dvostrano *(dual port)* višestruko baferovanje i čitanje podataka. Veličina bafera iznosi 128 sektora, podržavaju DMA ali ne podržavaju pisanje u režimu dvostruke reči.

Kvalitet rada je poslovično sjajan, kako smo već i navikli od ove kuće, kako u slučaju diskova, tako i u slučaju "Paradise" video kartica.

Quantum LPS serija

Svi modeli ove firme podržavaju dvostrano (*dual port*) višestruko baferovanje i čitanje podataka. Veličina bafera iznosi 197 sektora, podržavaju DMA, ali ne podržavaju pisanje u režimu dvostruke reči.

Probali smo ukupno četiri modela – LPS170A (170 MB), LPS270A (270 MB), LPS340AT (340MB) i LPS540AT (540 MB). Odmah napominjemo da se prvi model ubrzo povlači sa tržišta, pa ćete njegove primerke sigurno moći da nađete po veoma povoljnoj ceni. Model 170 ima 1011 cilindara, 15 glava i 22 sektora po tragu, model 270 ima 917 cilindara, 12 glava i 48 s.p.t., model 340 ima 1011 cilindara, 15 glava i 44 s.p.t., model 540 ima 1048 cilindara, 16 glava i 63 s.p.t.

Conner CFA540A

Ovaj disk ima dvostrano **(dual port)** višesektorsko baferovanje i čitanje podataka. Veličina bafera je 512 sektora, podržava DMA, ali ne podržava pisanje u režimu dvostruke reči. Brzina prebacivanja glave sa traga na trag je kraća od 15 mikrosekundi, što je jedino ovom modelu u celoj grupi pošlo za rukom.

Ovaj disk je ujedno i najviše "šminkerski" – iako je uobičajeno crne boje, na gornjoj ravni ima ubačen deo od anodizovanog aluminijuma. Računo da ima pomalo plani pažnju. Inače disk ima 1048 cilindara, 16 glava i 63 sektora po tragu.

SHOW TIME, FOLKS!

I tako složismo ih sve ispred sebe, razmontirasmo našu test mašinu i krenusmo na posao. Uzgred, naša test platforma koristi Intel procesor 80486DX na 33 MHz sa 256 KB keš memorije, 8 MB RAM-a i WD Paradise 90C31 ISA SVGA karticu.

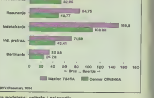
Podimo od porodice Quantum diskova. Odmah da kažemo da su njihova performansa veoma nalik jedne drugoj uprkos vidljivoj razlici u modelima u smislu tehnika. Tako modeli 170 i 340 vuku korene iz jedne, a modeli 270 i 540 iz druge generacije (to se vidi po vrstama kontrolera). Pa ipak, postoji jasne nit konzistencija kod svih modela, kao na primer u slučaju višesegmentnog bafera, koji je 50% veći nego kod drugih diskova, sa izuzetkom modela Conner.

U celini gledano, ovi diskovi su potvrdili tradiciju kuće da se uvek nalazi pri samom vrhu, a dosta često i da definiše sam vrh. Model 540AT je imao najbolje rezultate u brzini prenosa od svih testiranih diskova. Razlike u performansama između više modela firme Quantum su najverovatnije nastale zvog povećane gustine zapisa, čime se smanjuje broj fizičkih pokreta glava za čitanje i pisanje za istu količinu podataka.

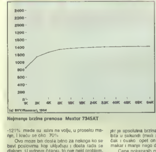
Interesantno je primetiti da je brzina prenosa podataka se i na disk u slučaju većih blokova (od 24 KB nagore) prelazila 2 megabita u sekundi, što je do pre jedva godinu dana bilo nezamislivo bez keš kontrolera. Dva modela troše po 4.17 VA, a druga dva (270 i 540) već solidnijih 9 VA. Uprkos toj danas već neobično velikoj potrošnji bili su mlaki na dodir posle 2+ sati rada.

Ni porodica WD diskova nije razočarala, naprotiv, i oni su osvetlali obraz. I u ovom slučaju bilo je razlika u performansama između dva modela, sasvim očekivano u korist većeg brata. Manji model 1210 je bio brži samo od najsporijeg diska u grupi, dok je veći bratić zauzeo počasno treće mesto po brzini. Manji brat troši skromnih 3.7 VA, a veći još skromnijih 2,97 VA, nije zato ni čudo što im treba set vremena da postanu bar mlaki na dodir.

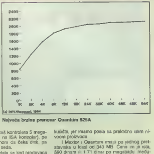
Maxtor 7345AT je zauzeo neslavno poslednje mesto u ovoj grupi. Prilično ubedljivo je zaostao za svim drugima, i to po svemu – i po brzini prenosa, i po radu sa bazama podataka. Istina, razlike u brzini u odnosu na najbrži disk u grupi nisu baš zanemarive i kreću se i do



Baza podataka: najbrže i najsporija



Najmanja brzina prenosa Maxtor 7345AT



Najveća brzina prenosa Quantum 525A

-121% mada su islini na volju, u proseku manje, I kreću se oko 70%

Ovo može biti dosta bitno za nekoga ko se bavi poslovima koji uključuju i dosta rada sa diskom U jednom čitanju, to nije neki problem, ali ako svaki čas pretražujete neku veću bazu podataka onda već imate ozbiljan vremenski problem sa čekanjem

I najzad tu je I kraj druma za ovaj put, Conner CFA540A Iako nije pokazao izuzetne rezultate u brzini prenosa, ovaj disk je imao najkraća vremena slučajnog pristupa (9 3 mS) što se jasno pokazalo kao prednost u radu sa bazama podataka

Ono što nema u apsolutnim vrednostima ovaj disk ima u konzistenciji jer je njegova kriva prenosa druga najravnija u grupi (iza Maxtor diska, koji u ovome apsolutno vodi iako je ta ravnina na nižem nivou apsolutnih vrednosti od drugih) Sa potrošnjom od 4 83 VA Conner nije ni najskromniji ni najalaviji posle par sati rada, bio je umereno mlak na dodir

ZAKLJUČCI

Prvo što moremo primetiti jeste da je opšti nivo performansi diskova veoma osetno porastao Na primer rezultati testova sa indeksiranjem pokazuju da su diskovi danas ravno dvostruko brži nego što su bili pre godinu ili dve dana i to je svakako veoma dobro Ovo dalje potvrđuju i krive brzine prenosa koja se danas u proseku nalaze na impresivnim nivoima – Maxtor je na dnu, sa prosekom od oko 1 3 megabita u sekundi, a Quantum modeli se približavaju proseku od 2 megabita u sekundi Sasvim je jasno da će već u narednoj iteraciji, koja će verovatno uključivati i veće keš memorije sve ovo otići dosta iznad 2 megabita u sekundi

Opšte performanse u poređenju sa referencom (kod nas popularnim diskom Maxtor 7213AT 213 MB) koja uzgred ima gotovo identična performanse kao i ovde prikazani veliki brat, model 7345A, su već sada otskočile, i to bogami poprilično Čak je apsolutni prosek dostigao zavidno visoka nivoe što sve nas može samo da raduje jer sada sa standardnim diskom čovek dobija brzina prenosa koje su pre svega godinu dana zahtevale keš kontroler Naravno kada sada Ia iste diskove povežete na onaj keš kontroler, ukupan rezultat je takođe pomeran nagore to ne treba da čudi jer je apsolutna brzina keš kontrolera 5 megabita u sekundi (misli se na ISA kontroler), pa čak i ovako opet on mora da čeka disk, pa makar i manje nego do sada.

Cene prikazanih modela se kod prodavaca kreću od 390 do 1 050 dinara. Najjeftiniji je Quantum LPS170A, sa cenom od 390 dinara S obzirom na performanse cena je čak odlična, a proizilazi da 1 MB košta 2 29 dinara

Sledi WD Caviar 1210 sa cenom od 430 dinara, odnosno 2,04 dinara po megabajtu Iako je megabajt jeftiniji (što je i normalno) u ovom slučaju su i performanse nešto odlije pa se stoga ovaj model može nazvati i boljim poslom od prethodnog

Sledeći su diskovi od 270 MB a to su Quantum LPS270A i WD Caviar 1270 Istog su kapaciteta praktično istih performansi i iste su cene 500 dinara ili 1,85 dinara po megabajtu Vaš izbor se svodi na proizvođača i dizajn kućišta, jer imamo posla sa praktično istim nivoom proizvoda

I Maxtor i Quantum imaju po jednog predstavnika u klasi od 340 MB Cena im je ista, 590 dinara ili 1 71 dinar po megabajtu međutim performanse su im sasvim različite Maxtor dosta daleko zaostaje za sličnim modelom Quantum-a, u proseku za 36 5% pa je i situacija jasna – preporučujemo Quantum LPS340A. Isto košta iz istog je čak izvora a radi brže

Najzad, visoka klasa Oba diska od 540 MB koštaju u disku isto nekih 1 050 dinara što daje 1 94 dinara po megabajtu Ovi diskovi su prikazali i najbolje performanse mada je Conner bio brži u proseku za nekih 1 5% (ovo je, doduše čisto akademska vrednost, ali i dalje stoji) Kapacitet je sasvim solidan izrada oba diska je superlativna, cena je ista – razlika u brzini je u korist Conner-a. Izbor je naravno vaš

Moramo ovde navesti jedan komentar koji smo čuli od dva velika beogradska prodavca Po njihovim nezvaničnim komentarima bilo je problema sa Conner diskovima prilikom rada pod *Windows* okruženjem Disk opisan u ovom tekstu smo zato probali i u radu sa *Windows*-om i moramo reći da nismo iskusili bilo kakav problem

Ovde prikazujemo rezultate samo najbržeg i najsporijeg diska uz podsećanje da najsporiji disk ima iste performanse kao i naša referenca Svi ostali se nalaze negde između ta dva ekstrema

Korisna adresa

ADA Computers
Tadeuša Košćuškog 72, 11000 Beograd
Tel 011 186 355
Fax/tel 011 186-267
(za diskove do 540 MB)

Korisna adresa

Geme Electronics
Mišarska 11, 11000 Beograd
Tel 011 332 275 ili 339-494
Fax 011 335-902
(za diskove od 540 MB)



Biosfera
Computer
2700
SVGA 512Kb mono
4Mb RAM + 256Kb cache
HDD 213Mb, FDD 1 44 ili 1.2
mini tower, keyb, mouse
2550
SVGA 512Kb mono
4Mb RAM + 256Kb cache
HDD 213Mb, FDD 1 44 ili 1 2
mini tower, keyb, mouse
2300
SVGA 512Kb mono
4Mb RAM + 256Kb cache
HDD 213Mb, FDD 1.44 ili 1.2
mini tower, keyb, mouse
1750
386DX/40
SVGA 512Kb mono
4Mb RAM + 128Kb cache
HDD 213Mb, FDD 1.44 ili 1.2
mini tower, keyb, mouse
moguće doplate za...
proširenja, kolor monitore, kućišta, egzotične komponente, modeme...
štampači...
EPSON matrični
HEWLETT-PACKARD laserski
PANASONIC telefoni
Dom Omladine, VI sprat, Makedonska 22, tel/fax: 3229-109, 3224-378, 3248-208

KAKO SU STVARALI ISTORIJU

Stari borci, opet ste u stroju! Sećate li se slavnih ofanziva iz 1984. i onih godina potom? Prilika je da se sumiraju rezultati i, kao na svakoj godišnjici mature, sagleda pređeni životni put. Da li ste sretniji, da li ste pametniji?

Jelena Rupnik

Nostalgije nikad dosta. Divna su to vremena bila. Svetlucanje potrošene mladosti. Sve je bilo slatko: komodori, spektrumi, džojstici, gumice... Ali ništa više nije kao što je nekad bilo.

Marks je pogrešno shvatio stvar. Istorija se možda prvi put javlja kao tragedija, a zatim ponavlja kao farsa, ali postoji i treći put – kada se istorija javlja kao *public domain*. Ovo pravilo najbolje se ogleda upravo u kompjuterskoj istoriji. Već su ispilvali računarski ekvivalenti onih koji su se sad opsetili kako su na sopstvenim leđima pridržavali binu dok je Bjelo Dugme sviralo na Hajdučkoj česmi. Čekam da iz nekog polprograma izmile i oni što su Bilu Gejtsu pridržavali miša dok je ovaj pisao MS DOS, plus kafanski kreativci koji su lično i jednoručno, u pauzi između dva špricera, oformili domaću kompjutersku scenu.

Postoji nekoliko proverenih načina za tretiranje istorije. Jedan od najboljih je poznat kao slavna prošlost ili „Gde si srećo? Mladosti?" Ko može da kaže da nikad nije zažalio za onim danima kada su programi u bejziku puštani u etar i prekucavani iz kompjuterskih časopisa?

Pravila za bolju primenu nostalgije: predmet nostalgije lakše se da podvrgnuti standardnom procesu nostalgizacije ako je u nekom prethodnom periodu bio rumen, veseo, šaren, uspešan, raspusan, a sada je bezbojan, ravan, hladan, dehidriran i ozbiljan. Izgleda da su računari skoro savršeni za nostalgiju.

PODGREJANA NOSTALGIJA

Nostalgija se lako deli. Čak i ako ste premladi da biste se sećali kako je to izgledalo raditi na drhturavim gumicama spektruma, dobro zagrejano društvo iz kluba programera očas će vam osvežiti pamćenje.

Nostalgija je najbolje kada je podgrejana. Ako imate problema da se prisetite šta vam je to sve prijalo u dobrim starim vremenima, podsećanje na nove ne tako dobra vremena će vam pomoći.

Postignut je ogroman napredak, a i dalje se retrogradno napreduje vrtoglavom brzinom u svakom pogledu. Vaša odluka da ne zastarite je zastarela još pre nego što ste je doneli. Ovo je gore nego sa ženskom modom, koja se menja od sezone do sezone. Gde su sada dežurni kočničari? Ni reakcionari nisu više što su nekad bili. Izmenili su ulogu – nekad su se od kompjutera branili, a sada pomoću kompjutera napadaju. Možda je računare ipak trebalo zatrti dok su još bili mali?

Srećom, mi smo ispred sveta fazno pomereni, i ponovo smo u situaciji da pre svih otkrijemo divotu prašine i prostih zadovoljstava. Oni tamo još ne znaju slasti toga, ali je samo pitanje dana kada će. Oni još uvek misle da je sreće srazmerna rezoluciji ekrana, broju megabajta, megaherca, mipsova i šarenih ikonica.

Naučenjaci kažu da bi ovaj svet sasvim drugačije izgledao da se neka od prirodnih konstanti (recimo gravitaciona) razlikuje u jednoj decimali. Desila se u međuvremenu mnogo gora stvar: narušen je zakon o održanju. (Da li se o tome još uči u školi?) Znate ono: održanje materije, održanje energije, održanje količine kretanja, održanje momenta količine kretanja, budala, vanzemaljaca, neprijatelja... Sa softverom to ne važi. Softver se u zatvorenom sistemu razmnožava bez granica. To stvara potrebu za povećanjem/pojačanjem hardvera, što još više utiče na širenje softvera. On je u stvari najveći virus.

Vredi priznati poraz. Pobedili su oni koji su zagovarali programiranje u mašincu. Pobedili su oni koji su mirisali nožice kompjutera. Pobedili su oni koji svojim hard diskovima daju ime i prezime. Pobedili su oni koji čitav pubertet provode na BBS-u. Pobedili su oni koji iz principa ne igraju igre (osim *Flight Simulatora*). Uzgred, kažu da tim koji pobeđuje ne treba menjati.

I šta sada? Svi se prave kao da je sve normalno. Nije da nisu mnogi posumnjali da nešto nije u redu: novi programski paketi sve su glomazniji, ali niko se ne usuđuje naglas da kaže da je koncept *Windows*-a jedan veliki corak. Naravno, ne sa stanovišta Bila Gejtsa i ostalih koji su zaradili lovu na tome. Ni heroji nisu više kao što su nekad bili. Nema više onog: kada je trebalo silnima reći, rekli smo odlučno: Ne! Možda to i nije tako čudno: od čega bi živeli razni kompjuterski tumači i objašnjavači novih tehnologija da je sve, ne daj bože, jednostavnije?



Ilustracija: Miša Marković

Vi prvoborci, koji ste bili tu u stroju pre deset godina, još uvek prisustvujete Času istorije. Je l' sada držim čas. Dobro, sve će to ponovo da se napiše, kako već biva. I kroz deset godina će vas, svedoke istorije, tagati novi svedoci. I opet će Nostalgičarsko društvo svirati u rog (*St. Pepper's Nostalgic Hearts Club Band*). Zato retrospektivne nostalgične poeme i hvalospeve vremenu prošlom velikodušno prepuštam drugima.

Pre nego što krenete na spavanje, obiđite svoje mrtve i zaboravljene komodore i spektrume na tavanu ili plakaru, obrišite prašinu i odigrajte još jednom *Manic Miner*-a. Što se mene tiče, uvek sam više cenila *Impossible Mission*.



PONOVILO SE NA DANAŠNJI DAN

U proteklioj deceniji ne računerskoj sceni u našoj zemlji se menjelo i promenilo puno toge. Jedino je odnos prosvetnih vlasti preme računerime ostao isti. Nevenke Spalević, svojevremeno urednice obrazovnog bloke "Pet Plus", ni nekon toliko godine neme nijednu lepu vest za vas.

Nevenka Spalević



Sudeći po izjavi koju je dao E. Dijkstra po povratku iz posete Sovjetskom Savezu kasnih sedamdesetih godina, Amerikanci smatraju svojom najvećom pobedom u hladnom ratu to što Rusi proizvode IBM računare. Vodeće sile u proizvodnji i upotrebi računara stiču prevlast u skoro svim oblastima ljudske delatnosti, jer upotreba računara obezbeđuje nedostižno povećanje produktivnosti u odnosu na tradicionalnu organizaciju rade. Što se čovek više radno engazuje ima veću potrebu za alatom koji bi povećavao efikasnost njegovog rada. Računari nude fantastične pogodnosti, to je jasno svakom obrazovanom čoveku ali „naši ljudi u praksi nemaju dovoljno probleme za njih, jer uglavnom malo rade. Pošto malo rade, imaju i malo probleme i mogu da ih rešavaju i bez računara" govorio je na stranicame našeg časopisa 1985. godine dr Radomir Mihajlović. U to vreme, kada su počeli da izlaze „Računari", i u našem društvu je sazrevala svest da moramo da uvažavamo tvrdnju svetskih priznatih ekonomista kako je kompjuterizacija prva stvar kojoj zemlje u razvoju moraju pristupiti ako žele da uhvate korak sa razvijenim. Tog ozbiljnog posla prihvatile su se mnoge male zemlje i iskoristile svoju šansu u trećoj industrijskoj revoluciji, za razliku od naše koja razvoju računerstva i informatike nije pridavala značaj koji zaslužuju.

"NEVEROVATNE ČINJENICE" NEKAD I SAD

"Najviše zabrinjava neverovatna činjenica da ne postoje efikasne društvene institucije koje bi u SR Srbiji bile zadužene za brigu za razvoj računarstva i informatike" rekao je profesor računarske tehnike i informatike na beogradskom Elektrotehničkom fakultetu dr Jozo Dujmović u intervjuu datom pre devet godina za naš časopis („Računari" 5/1985). Činjenica koja je bila neverovatna za našeg uglednog sagovornika, čiji je projekat za izbor *Data Base Management* sistema pobedio u velikoj konkurenciji na konkursu koji je za potrebe agencije američke vlade raspisao *National Bureau of Standards* iz Vašingtona, tačna je, na žalost, i danas. Što je još gore, uneslo da se sada, kada preživljavamo duboku ekonomsku krizu, više bavimo informatičkim obrazovanjem u školi i njegovom primenom u svakodnevnom radu, [illegible] školuje se deca po nastavnim programima u kojima broj časova iznosi manje od 17% od vremena koje je pre desetak godina bilo posvećivano upoznavanju mogućnosti računara.

TANGO PROSVETNIH VLASTI – KORAK NAPRED DVA KORAKA NAZAD

Te 1985. godine najavljena je nova reforma obrazovanja kojom je dat bolji tretman informatičkim sadržajima u nastavnim programima. „Ideja informatičkog obrazovanja, čiji su značaj konačno svi uvideli, svoz je velsi u jednima nove reforme. Ovog meseca na javnoj diskusiji naći će se nastavni planovi i ako sve bude teklo kako je predviđeno, već u februaru ćemo znati šta će se učiti u novoj školi od jeseni. Izvesno je da će svi učenici prvog razreda u okviru predmeta Osnove tehnike i proizvodnje sticati informatičku pismenost, dakle informatičko obrazovanje tretiraće se ubuduće kao element opšte kulture" – pisali smo u decembarskom broju 1985. godine. Smatrali smo da je uvođenje opšteg informatičkog obrazovanja bio značajan korak napred našeg školstva i našeg društva.

"Ali glavni problemi nisu u samim nastavnim programima, već u načinu njihova realizacije. Daleko od toga da su materijalne mogućnosti našeg društva dobre, ali ako već možemo da se prihvatimo organizacije Olimpijade (u Sarajevu 1984) jer računamo da će nam se ulaganja vratiti, zar još ne uviđamo da se ulaganje u obrazovanje vraća? Konačno, vreme je da se upitamo kako ćemo živeti za deset i dvadeset godina kada budemo [illegible] od onih u koje danas treba da ulažemo.

Možda [illegible] pravi problemi u realizaciji programa iz informatike i računarstva bili presudni da u novoj reformi obrazovanja pet godina kasnije informatika praktično bude izbačena iz gimnazijskih programa. Naime, prema danas aktuelnim nastavnim programima, broj časova informatike i računarstva koji je u prethodnom obrazovnom sistemu iznosio i do 780 (za učenike zanimanja matematičko-programerski saradnik) smanjen je u sadašnjim gimnazijama na (72–80). Ovo je se pribeglo kako bi se napravilo više mesta za predmete koji daju opštu kulturu" (?), među potrebama za informatičkim obrazovanjem danas nije smanjena, naprotiv. Ako je nekome bio cilj da na taj način spreči odliv mozgova, u tome će verovatno uspeti, ali ćemo ispuniti sami ostati bez danas u svetu najtraženijih stručnjaka.

SVET DRUMOM A MI ŠUMOM

Procena autora tekućeg obrazovnog sistema da informatika treba da se uči kroz sve četiri godine u ekonomskim školama, a samo u prvoj godini gimnazija [illegible] tehničkog usmerenja [illegible] predstavlja grubu [illegible] društva [illegible]. Posle ovakvog odnosa prema informatici, učenici gimnazija se u startu daju manje šanse ne samo za uključivanje u rad već i za korišćenje [illegible] informacionih tehnologija, što objektivno umanjuje njihovu sposobnost da prate dostignuća u bilo kojoj naučnoj disciplini.

To što je gimnazijskim programima predviđena fakultativna nastava iz ovih disciplina ne olakšava [illegible] okolnost da se fakultativna nastava informatike izvodi samo za decu čiji roditelji mogu da finansiraju ovaj oblik obrazovanja. Polazeći od značaja informatičkog obrazovanja za razvoj društva, rezultata ankete o zainteresovanosti učenika gimnazija za učenje informatike u okviru redovne nastave i posle prvog razreda, iskustava iz prethodnih reformi obrazovanja i iskustva iz škola u kojima informatički sadržaji nisu tako drastično [illegible] kao u gimnazijama, neosporno je da su [illegible] odnos fonda časova informatike i potreba za informatičkim obrazovanjem u redovnoj nastavi u gimnazijama u velikom nesk[illegible].

Na ovakav tretman informatike u gimnazijskim programima, koji nas skreće sa glavnih civilizacijskih tokova, [illegible] malo rezultata koje smo postigli uvođenjem informatike u redovnu nastavu još 1979. godine, a čitave generacije mladih okreće prošlosti umesto [illegible], reagovali su u više navrata Društvo matematičara Srbije, Matematički fakultet iz Beograda, Matematički institut SANU i Institut za matematiku PMF-a [illegible] u Novom Sadu, Aktiv srednjoškolskih profesora Beograda i mnogi drugi. Kao što vidimo, tu nije bilo poznato ili nije bilo od značaja za prosvetne vlasti koje su usvajale nastavne programe. Možda bi sve bilo drugačije da nije bilo napred navedene „neverovatne činjenice".

U vreme kada kompjuteri postaju produženo ja savremenog čoveka, kada komjuterski vođene rakete [illegible] ratove [illegible], kada nema oblika proizvodnje [illegible] u kome se intenzivno ne koristi računar, srpska deca su lošim i nekompetentnim [illegible] – a možda i lošim i nekompetentnim [illegible] intervencijama prosvetnih vlasti osuđeni na računarsku nepismenost. [illegible] nalaz po našim školama je bio zastarelo već u momentu kada je nabavljeno. Nastavni programi jednostrano ne odslikava sadašnje stanje [illegible] iako je usvojen maja 1990. [illegible] pisale su pomenute stručne i naučne organizacije [illegible] zahtevale da se stručna pitanja vezana za nastavu računarstva u srednjim [illegible] školama, što se pre svega odnosi na povećanje fonda časova i kvalitetno [illegible] nastavnih planova, odmah predaju [illegible] kompetentnim [illegible] i da se [illegible] koje se predlože uključe u nastavni program.

Ministarstvo prosvete Republike Srbije do sada nije reagovalo. Da li [illegible] da je sve informatičko obrazovanje samo privilegija dalekovidih roditelja dubljeg džepa [illegible] modernog organizovanih [illegible] društva u celini? Ili je [illegible] da je [illegible] oblasti informatičkog obrazovanja samo posledica ukupne krize društva?

Stvar je [illegible] informatičko obrazovanje svoj [illegible] okvire u kojima se rešava. Sve [illegible] informatike i računarstva ne podigne na nivo [illegible] izlaz iz informatičkog mraka [illegible] može se sagledati.

Krajnje je vreme da se [illegible] pamet. Otpočnimo ukidanje [illegible] blokada u sopstvenim glavama. Realan [illegible] deo deklarativnog izjašnjavanja za naučno-tehnološku revoluciju, informatičko društvo [illegible] privrede i uključivanje u razvoj [illegible] sveta. Kako stvari trenutno stoje, imamo perspektivu tehnološkog [illegible], jedino što se vodeće ekonomske [illegible]. Ako nam uloga priveska ne odgovara, treba da eliminišemo „neverovatne činjenice" [illegible] da kompletan koncept informatičkog obrazovanja [illegible] predškolskih ustanova do poslediplomskih [illegible] ljudi od struke. Takvih ljudi koji [illegible] kako treba raditi da se podigne nivo opšte kulture [illegible] računara, njihovom [illegible] napredak ima dovoljno u našoj [illegible] jedinu [illegible] šansu da rade. Pameti [illegible] ne nedostaje, jedini je problem što više [illegible] da je sačuvamo umesto kako da je upotrebljavamo.



prOcesor koji Vam nedostajE
BS Procesor
BEOGRAD, Hodži Nikole Živkovića 2, tel/fax: 011/180-739, 632-162, 632-882 tel 634-548 3281-321(404
art studio CRNO NA BELO

NEKA DRUGA STVARNOST

Kiberpank, računarsku potkulturu, jedni vida kao „tehnologiju sa duhom", za druge je to „fuzija čoveka i mašine", za treće, opet, „sveta alijansa avata tehnike sa podzemnom pop-kulturom i uličnom anarhijom". Bilo kako bilo, ova mešavina opsesije visokom tehnologijom i sarkazma nad njenom konvencionalnom upotrebom danas je gotovo svuda oko nas.

Kada momak iz Japana tehno di-džej Bejbi Tokio „zakači" na Internet poruku „Ljudi upotrebljavaju tehnologiju za rat. Želim da upotrebim tehnologiju za muziku, umetnost i za život," a ovdašnja deca, vična hakingu („dale autorirada" više ne prolazi ovde) je „skinu" kao sebi birsku – kontrakultura kompjuterske ere pokazuje svu svoju žilavost spram policajaca informacija (datacops) gospodara rata, ksenofobičnih komšije rigidnih matoraca i ostalih poklonika dirigovanog društva – i „tamo" i „ovde".

"Oni rade na tome da nas pretvore u pečurke drža uas u mraku i hrane nas govnima!" žesti se jedan kiberpanker sa Vračara.

Đavolski blizanac Utopije – Distopija – to imaginarno kiberpankersko depresivno bedno mesto u kome ljudi žive grozan život – za mnoge je film koji curi uživo. „Voleo bih da se vratim kući u Sarajevo da mi ulšta ne bude srušeno da opet imam kompjuter i kućnog ljubimca," kaže jedan tinejdžer „sa ratom zahvaćenog područja".

"Sve to ima neku osnovu u današnjim tehnologijama," objašnjava Pol Safo istraživač na Institutu za budućnost u Menlo Parku u Kaliforniji. „To su mladi ljudi čiji pokret anticipira ono što će doći koji shvataju da mogu promeniti budućnost a mi ostali ćemo živati u njoj."

Hauard Rajngold američki autor misli da kiberpank pokreću mladi ljudi jer žele, s jedne strane da naprave pokret za koji mogu reći da je samo njihov a sa druge strane jer im je dosta starih čudaka koji se svima penju na glavu pričama o tome kako je nekad bilo dobro.

Ovako ili onako, nova potkultura koja se kompjuterskih ekrana šiiem sveta. Taj futuristički hiperspejs jedni vide kao „tehnologiju sa duhom" (Stjuart Braul izdavač) za druge je to „fuzija čoveka i mašine" (Rudi Raker matematičar). Za Brusa Sterlinga SF pisca kiberpank je „sveta alijansa tehničkog sveta sa podzemnom pop-kulturom i uličnom anarhijom".

Kiberpank (*cyberpunk*) je izraz s kraja dvadesetog veka dobijen od reči *cybernetics*, ili nauke o kibernetici teoriji upravljanja i *punk* što mu dođe kao pobuna protiv društva. Ova mešavina opsesije visokom tehnologijom i sarkazma nad njenom konvencionalnom upotrebom omogućava svakom „kiberu" da drži glavu iznad vode košmarnog sveta na koji ne može da utiče aktivno učestvujući u novom medijumu čija virtuelna realnost ne samo da je otvorena i podložna istraživanju već se njome može i manipulisati.

KIBERPANK MIŠOLOVKA

Virtuelna stvarnost je danas skoro svuda oko nas. U SAD gde je najjača kiberpank mreža nije ulita neobično da se samo u nekoliko dana promoviše nova Segina video igra *VirtuelRacer*ČMDNMĆ Spectrum HoloByteova *Star Base One* i *Visions of Reality* firme Cybergate. Istovremeno muzej Pegi Gugenhajm u Njujorku predstavlja VR izložbu sa radovima Dženi Holcer i kompozitora Tomasa Dolbija, dok se u sajmu američka armija prikazuje VR simulator tenka. Kiberpank imidž vrela naučnofantastičnog nadrealizma uvrnutog kompjuterskom grafikom našao je put do umetničkih galerija muzičkih videa i holivudskih studija. Kiberpank magazini obavezno distribuirani elektronskom poštom umnožavaju se kao kablovska televizija. Popularnost kiberpank muzike je tolika da nekoliko produkcijskih kompanije živi samo od nje. („Voliš li tehno?" – pita me skoro jedan beogradski gimnazijalac sa kiberpank nadimkom Dominator.) Tome svakako treba dodati i podatak da se kiberpank knjige, poput „Zabranjene beogradske bibliotekarke verovatno ne mogu čudom da se načude gde nestadoše primerci „Neuromansera". Naravno pokupili su ih ovdašnji kiberpankeri.)

Vesna Ćosić

Tokom više od jedne dekade virtuelna realnost pomagala je pilotima na obuci da grafički vizuelizuju objekte u tri dimenzije i polju od 360 stepeni. Napredak u smanjenju veličina i troškova hardvera i softvera izvodi virtuelnu realnost iz ezoteričnog sveta simulatora leta i stavlja je na dohvat arhitektima, dizajnerima, hirurzima i – deci. „Ako se to valjano uredi virtuelna stvarnost ima priliku da izmeni svet," kaže Dejvid Bonini, direktor firme Division Inc. čiji sistemi virtuelne stvarnosti pomažu istraživačima lekova da asembliraju virtuelne molekule.

Druga strana medalje od koje se s razlogom ježe roditelji i pedagozi je to da se u kiberpank i VR žurma ima hvala dobri efekti psihodelika i veštačkih droga jer su „super za razmenu informacija u mozgu". Aktivna publika kiberpank medija sastavljena od hakera futurista, marginalnih naučnika i kompjuterskih umetnika – gute lazone tipa „da je moguće naći prečicu kroz vremenski tok" da veštačka droga centrofenoksin „uvećava inteligenciju" a hiderdžin „povećava mentalne mogućnosti". Kiberpank magazin *Mondo 2000* čiji urednik R. U. Sirijus voli da se hvali intervjuima sa poznatim ličnostima i informacijama o tome šta je „vruće" a šta je „hladno" – ima za moto stav „da inspiriše i instruiše ali ne i da vodi". Ali kada se u paketu kontrakulture kompjuterske ere nađu i trabunjanja da je „bezbedan seks dosadan" a „jeftina uzbuđenja zabavna" zabrinutost odraslih lako prelazi u paniku. Naravno mnogo toga nije drugo do kiberpank poza.

"Konačno se," kaže Ben Dilejn izdavač publikacije *CyberEdge Journal* „tehnologija našla pred izazovom. Mislim da ćemo videti virtuelnu realnost svuda oko nas. To je ona bolje mišolovka i bolji način rada sa kompjuterima." Lepo lice kiberpank pokreta jasne stavlja do znanja da je po definiciji globalan univerzalan futuristički nefasciniran istorijom da nije rigidan u odnosu na razlike (naprotiv podrazumeva ih) i da je pacifistički. „Istorija je zanimljiva stvar za kiberpank," kaže Kristofer Mejer dizajner, „jer to su sve podaci. Međutim istorija zahteva dosta prostora na disku a mnogo od toga je čista buka."

IGRALIŠTE ZA HAKERE

Džon Lata predsednik firme „4th Wave" za istraživanje tržišta predviđa da će se nevojna industrija virtuelne stvarnosti uvećati za oko pet puta u sledeće četiri godine. Uostalom kiberspejs čija je centralna uloga u kiberpankerskom pogledu na svet uespoma, prenosi milione glasova faks i kompjuterskih poruka preko globalne žukrštene telefonske mreže. Kiberpank prostor je više od igrališta za hakera. To je novi medijum. Svake noći na *Prodigy CompuServe GEnie* i desetinama drugih BBS-ova armije ljudi okupljaju se na kompjutersko ćaskanje na interaktivne debate koje ne priznaju zidova vremena nacionalnosti i socijalnog statusa.

Ništa neobično onda što su se veliki i mali preduzetnici uhvatili za virtuelnu realnost u nadi da će pretvoriti tehnologiju ministarstva odbrane u unosan šou biznis. Kompanija na prostoru od reke Hadson do Tokijskog zaliva među kojima su i Paramount Communications AT&T Viacom Sega Nintendo Sony Matsushita Edison Brothers Hasbro i Time Warner – očekuju kumulativne milijarde dolara od virtuelne realnosti. Kristofer Džentile iz firme Abrams/Gentile Entertainment koja razvija VR sistem za domaćinstvo u Prinstonu Nju Džersi predviđa snažan prodor 1996 godine. Šta mislite o 3D televiziji? Kupovini pomoću VR? Seks mreži za domaćinstva? „To postaje potencijalna erupcija," ubeđen je investitor u području medija Marti Pompadur.

USPAVANKA ZA ČULA

Tipično kiberpank „dobro mesto" izgleda dovoljno ugodno puno je raznih signala jednom rečju uspavanka za čula. Patinirane fotografije avanture iz ranijih vremena rana izdanja Žila Verna i Dorijana Greja bar od mahagonija gde se služe pametna pića – poželjna su „scenografija". Ako uđete u istraživačevu oblaju prednju prostoriju radnje Virtuelnog Sveta u Volnat Kriku (*Walnut Creek*) videte da se iza montažnih zidova pupravna nešto čudovišno. Pal do ludila dovedenih fanatika spremnih za ubijanje će uskoro stupiti u smrtonosnu bitku u stilu 21. veka protiv novajlija sa lukavim kompjuterskim nadimkom *Cyber Rick!*

BattleTech desetominutna interaktivna video ekstravagancija baca vas u ogoljeni svemirski pejsaž pesul monolitima sa zadatkom da neutrališete četu teško uočljivih robota – vaših protivnika. Proces se naravno zove Virtuelna Stvarnost. Firma „Virtual World Entertainment" koja pored svojih prostorija u Volnat Kriku ima podružnice u Čikagu Tokiju i Jokohami je samo deo ovog brzo rastućeg spoja umetnosti nauke i spektakularnih efekata. „Želimo da kažemo šta je sve moguće," poručuje grupa ekscentričnih kiberpank vizionara nastanjena u ruševnoj viktorijanskoj kući sa pogledom na brda u Berkliju iz koje je izašlo delo „Vodič do Novog Ugla".

Danas kiberpank VR zabava cveta tamo gde je i film počeo pre skoro stotinu godina – u oblasti arkada. Stavljanje novčića u proraz mašine nekada je nudilo mogućnost da se zaviri u tajnovito mesto gde Fatima pleše. Sada 4 do 30 dolara vodi u ludu vožnju svemirskim brodom (*Cybergateu*) ili zastrašujuću šetnju kompjuterizovanim Akropoljem (*Dactyl Nightmare* firme „Virtuality"). Nije tajna: video igra su biznis od 5.3 milijardi dolara samo u SAD skoro iste veličine kao bioskopsko tržište. Kiberpank filmovi a la *Blade Runner Videodrome Robocop Total recall Lawnmower Man* i naravno *Terminator 2* pomerili su se sa kultnog tržišta na glavnu struju. Mladi kiberi u Japanu odlepili su na film po imenu *Akira*. To je sci-fi moralistička priča u kojoj kiberpank banda vodi borbu po ulicama razorenog Tokija posle trećeg svetskog rata – i pune kase zadovoljnih investitora.

KOMPJUTERSKI ŠLEM

U doslednu definiciju virtuelne realnosti ne mogu stati baš svi efekti. Kiberpank poklonici govore o mreži kao da je „virtuelne" dekie moguća, prikrivena realnost koje može biti otvorena istražena i na koju se može uticati. Svi koriste kompjuterski šlem koji dopušta pomeranje objekata u sajber prostoru i mnogo elektronskog zezanja. Međutim neki kiberpankeri „idu" u VR prostor ne samo da bi ćaskali i da bi se igrali: neki odlaze da u njemu žive i da umru. Pre tri godine jedna od najmodernijih elektronskih zajednica na Internetu – WELL (*Whole Earth 'Lectronic Link*) – bila je žestoko uzdrmana kada je jedan od najaktivnijih korisnika nekoliko nedelja posle „virtuelnog" izvršio istinsko samoubistvo potresen činjenicom da kada god uključi kompjuter ovej mu obriše svaku poruku koju je iko ikada ostavio. Jedan novobeogradski osnovac kada mu je otac na kratko zabranio prilaz modemu i kompjuteru rekao mi je sasvim ozbiljno: „Znaš mislio sam da bez modema ne mogu da živim. Eto mogu. Ali teško."

Većina stvari za kojima luduju kiberpankeri je u stvari virtuelno virtuelna varijacija na temu arkadnih igara koja lukavo smenjuju kretanje kroz vreme sa uzbudljivim slikama na ekranu. Tome treba dodati da svi sistemi imaju zajednički cilj da omoguće nove horizonte, da pipnete i osetite da pruže uzbudljivo iskustvo.

Virtuelna stvarnost? Hm možda. Virtuelna teatralnost? Sasvim izvesno! Omni-okreni kada je daleke



čevanje borilačke veštine) Ove „igračke za dečake" pravljene su za kompjutersku generaciju. Koje teško podnosi „ljude iz drugog filma". Zato je valjda skoro nemoguće videti VR igru na bazi nekog sentimentalnog filma za starije građanstvo – naravno ako to nije *Driving Miss Daisy Crazy* sa staromodnim autićem koji juri sporednim putevima pred poterom negativca.

S razlogom. Film i TV su pasivno iskustvo. VR igre su interaktivne. I što ste aktivniji to dublje u njih ulazite. Igrači koji visoko cene svoje veštine ubijanja i razbijanja razvijaju snažno motivisanu profesionalnost: oni postaju posvećenici koji se stalno usavršavaju, dok trgovci postaju sve bogatiji. Liberalniji analitičari se slažu da VR može biti i medijum socijalizacije čak i kod one „pal-mrtav si" vrste. TV, video igre i video kasete drže ljude u hermetičnoj izdvojenosti. VR ih izvlači iz kuće pomoću nove smicalice – po ideji mamca koji su osmislili proizvođači filmova tokom pedesetih godina ovog veka, suočeni sa izazovom televizije i ponudili publici sklonoj da ide u bioskop automobilske trke, 3D kopije i rakete u tehnici Sinerama. VR omogućava da po volji luta kroz polja sila ili minska polja, nudi demokratiju zabave. VR programer Rendal Valser je to ovako definisao: „Onaj ko pravi film kaže: Pogledaj, pokazaću ti! Onaj ko pravi sajber prostor kaže: Ovamo, pomoći ću ti da otkriješ!"

A kojih sve tu svetova za otkrivanje ima! Mada su neke igre zastarele već u vreme kada se nabavljaju i mada likovi mogu biti razrađeni tek na nivou *Space Invaders*-a – one zaista daju osećaj da ste tamo negde, bez težine i bezbrižni u tom tajnovitom mestu između ušiju. Neke od avantura su potpuno blesave. U *Virtual Adventures* firme „Iwerks Entertainment" klizite pod vodom da biste izbavili neka retka jaja koja je položilo benevolentno biće iz Loh Nesa.

59. godine futurista Vilijam Burou opisao izgubljenu zemlju kiberpank žanr nisao je u proceduru s kult. teme Interzone u srca i rečnik kiberskih entuzijasta i umetnike.

„Surfuj ivicom!" – galame mladi Ameri jurcajući kiberpank prostorom. Ovdašnja deca, kad se otisnu u kiberspejs, prvo istražuju „šta ima novo" u svetu igara, nauke, muzike, umetnosti. Pomoću kompjutera i modema „listaju" elektronske časopise, čitaju elektronsku poštu, „skidaju" igre, kopiraju softver. Jer, umreženi kibernetički prostor jeste magnet za nove mislioce ko je stotine supkulture, kultova i životnih stilova inspiriše, samo ne – kreativne interaktivne debate.

„Moji roditelji kad ne govore o ratu, svađaju se ko je prvi počeo. Noću sanjaju kako da nabave brašno, ulje i šećer. Mozak im se pretvorio u puding. Izbrisali su im sve vijuge, sem onih za strah, bes i preživljavanje. Ja sam iz drugog filma," kaže jedan kiberpanker iz moje ulice, koji svake večeri, ne izlazeći iz svoje sobe, „odlazi" na kompjutersko ćaskanje po beogradskim BBS-ovima.

„Data autostrada", globalna komunikaciona mreža, na čiju je infuziju prikopčano dvadesetak miliona zavisnike što više ispred svojih kompjuterskih monitorafeltara – verujući u spas uz pomoć personalnih računara – hrani sa slobodom informacija.

FANTOMSKI RIBNJAK

Kada je u godini Zmaja, dakle 88. talentovani komelovac Robert Moris opačio Internet, eksperimentišući sa kompjuterskim virusima, aslilom telekomuteri, kiberpankeri i ostali stanovnici računarske interzone pali su u očajanje. Domaći ljubitelji bitova i bajtova, šta cure umrtvljenom planetom žive svoj zatvorski košmar, ne prestajući na komunikaciono samoubistvo. Modemski hakeri novi su (anti)heroji, čija veština provlačenja telefonskom žicom „bez pasoša" donosi glasove šta je „vruće" i gde su horizonti. „Trulo mi je da živim u mraku. Ne mogu da čekam da prođu sankcije da bih našao rešenje za moj programerski problem. Prijatelj mog prijatelja je profa u Kanadi. U univerzitetsku mrežu ušao sam lako. Znao sam ime njegovog psa. Ljudi u ogromnom broju slučajeva uzimaju za šifru ime kućnog ljubimca. Imao sam sreće. Logovao sam se bez problema. Džukac je upalio," kaže meloletni programer iz Beograda.

Kada je Brajen Silverman bio student na MIT u, sagradio je desel stopa visok kompjuter od sklepanih delova i pecaroškog konca. Danas, kao projektant softvera pri kompaniji „Logo Computer Systems" u Montrealu, smislio je novu vrstu računara: virtuelni kompjuter koji nastanjuje ekran običnog PC-ja. Silvermanov program vam omogućava da projektujete vaš sopstveni virtuelni računar i potom posmatrate signale kako putuju kroz logičke kapije. Što će reći: informaci je bivaju procesirane pred vašim očima. Program je nazvan „Fantomski ribnjak" (*Phantom Fish Tank*). „Gospodar ćelije" (*Cell Master*) grafički je savršenija verzija, čiji kreator Mark Bolme ima običaj da kaže: „Tek smo zagrebali površinu onoga što se može učiniti sa ovim stvarčicama."

IGRAČKE ZA DEČAKE

Transkulturalno pleme kiberpankera nema volju ni nameru da govori ljudima šta i kako da misle; cilj je pokazati kakve su mogućnosti i, naravno, biti kooperativan. Košmarni svetovi koje vrte kult autori kiberpank umetnosti razvijaju varijante zloupotreba i kroz uznemirujuće slike simbiotičke veze ljudi i kompjutera gradači viziju budućnosti koja ne trpi ravnodušnost. Ništa neobično onda što i Marvin Minski, otac veštačke inteligencije, vidi kiberpank pokret kao „najozbiljniji pokušaj da se shvate posledice i implikacije pojedinih stvari".

Ako su koreni kiberpanka isto toliko literarni koliko i tehnološki, onda je igra izvor nove imaginacije. Uz ogradu da su igre toliko nove da nema konsenzusa kako ih nazvati. Naravno, baza su u suštini kompleksne verzije ili porodična filmska žanra i formata video igara: svemirske avanture tipa Zvezdani ratovi (gađanje stvari), trke, vožnje tipa Olujni dani (vožnja preko raznih stvari) i boks mečevi tipa *Rocky* (udaranje ma

Iz Neuromansera

„Momče, to je gadan komadić softvera. Najluđa stvar od kako je prase zavrnulo rep. Ta prokleta stvar je nevidljiva. Logika je da se neprimetno prišunja meti i počne da mutira, tako da postane potpuno isti kao napadnuti led. Tada se mi priključujemo i upadamo u glavni program, zamajavajući logiku leda. Postaćemo sijamski blizanci pre nego što se uznemire." Fletlajn se nasmeje. „Voleo bih da danas nisi tako veseo, čoveče. Taj tvoj smeh mi struže po kičmi." – reče Kejs. (Vilijam Gibson: *Neuromanser*, kult delo kiberpanka)

BAL SVETOVA FANTAZIJE

Za sada prava VR zabava ostaje mala ekskluzivna delatnost: ona ne može da premešta toliko tela na sat koliko Zvezdane Staze mogu. Međutim, tu ima zabave za sve: potrošače i proizvođače. I jedni i drugi čine kiberpank pokret. Bal prema elektronskoj zajednici WELL, baziranoj u Severnoj Kaliforniji, gde je pitanje postoji li ili ne takav pokret izazvalo lavinu odgovore i višemesečnu debatu.

„Velika stvar u vezi sa virtuelnom realnošću jeste da smo mi sada u poziciji da postavljamo pravila," kaže Toni Eš, predsednik „StrayLight Corp.", dizajner uzbudljivih avantura *Cosmic Debris* i *Bank*. Neke tajna kompanije sa značajnom specijalizacijom u području vojnih VR aplikacija sada stiču nova iskustva kao učesnici u kreaciji igara. „Godinama smo bili u biznisu koji je zahtevao kreiranje modela realnih svetova," kaže Džef Edvards, marketing menadžer firme „Evans & Sutherland" koja je u saradnji sa „Iwerks"-om razvila *Virtual Adventures* i dodaje: „Sada smo u biznisu kreiranja svetova fantazije i naši kreatori imaju svoj bal!" No, ako pitate kult-autora kiberpank žanra na šta se odnosi *Neuromanser*, taj manifest kompjuterske supkulture, Vilijam Gibson hladnokrvno izjavljuje: „Za mene je važno da se *Neuromanser* odnosi na sadašnjost."

Ovako ili onako, *Neuromanser* i sve što uz njega ide može se shvatiti i kao metaforična evokacija današnjih tehnoloških tokova. A ono što tako neodoljivo privlači kibere planete ovom delu svakako je i Gibsonov košmarni opis imaginarnog sveta, uznemirujuće sličan onome koji oni nastanjuju. Junak romana, drogiran, sagoreli muvator po imenu Kejs živi u Interzoni, izgubljenoj zemlji u kojoj je sva tehnologija pretvorena tehnologijama podzemlja. (Opsesivna tema kiber stvaralaca još od Vilijama Borouza i „Golog doručka".) Interzona je, naravno, na ivici superkorporativnog globalnog sela, gde se Kejs bori sa Moli, lepoticom u čije su očne duplje implantirane refleksivne naočari, a prsti naoružani žiletima koji rade kao nož-skakavac. Misteriozni poslodavac kontroliše Kejsa i Moli, omogućavajući junaku dala da otvori kiberspejs, ili kibernetički prostor, izraz koji je izmislio Gibson. To, naravno, nije sve. Heroji Neuromansera ustvari rade za veštačku inteligenciju, koja pokušava da prevaziđe činjenicu da su kompjuteri još uvek pod kontrolom.

Bili za ili protiv kiberpanka, kreativno iskustvo ovog veka verovatno će ohladiti euforične percepcije tehnologije, a proizvodi virtuelne realnosti mnoge stvari učiniti jasnijim, ma koliko to paradoksalno zvučalo: pre svega, preteći naša ponašanja u fikcije koje se obraćaju modernoj svesti (i savesti).

MEKOVIH DESET GODINA

Svetom personalnih računara suvereno vlada IBM PC koncept, a za takvo stanje stvari najviše možemo zahvaliti pravom moru proizvođača koji su sa uspehom klonirali i unapredili ovaj standard. Ipak, ako bi trebalo reći koja platforma je najviše učinila da „personalci" postanu ono što su danas, odgovor je samo jedan – *Apple Macintosh.*

Milan Bašić

Te 1984. godine igrao sam se na malom *Sinclair ZX 81* ličnom računaru, a programe pisao „bušeći" kartice za *IBM 1130* ili kucajući na terminalu jednog *DEC PDP 11* kompjutera. Ipak, san mi je bio pravi „personalac" poput „Apple II". Njega je imao moj kolega sa fakulteta i kada bih ponekad svratio na kafu kradomice sam pogledavao na tu „strašnu spravu". Jednog dana, nekako u ovo vreme, prijatelj mi reče: „Prodajem računar, treba da mi stigne novi". Priznajem, mom iznenađenju nije bilo kraja. Zaustih da ga pitam je li u pitanju moćni *IBM PC*, no odgovor me preteže. Tada prvi put čuh za *Apple Macintosh*. Mesec dana kasnije dođoh na prvo viđenje. Vertikalna kutija, maleni ekran, nišla veća tastatura i miš bili su jedine stvari na stolu. Ne bih li sakrio zavist, naoko nonšalantno odgledah prezentaciju i rekoh:

„Računar kojim se upravlja mišem nije prava stvar. To je za Amerikance koji ne mogu da shvate ni zašto je 2 + 2 jednako četiri!"

Eh, u kakvoj sam zabludi živeo. Danas, deset godina kasnije, guram miša po stolu „sva u šesnaest", bez grafičkog interfejsa i mnoštva boja život mi je težak, a pisanja programa prihvatam se samo kada moram. Na žalost, još uvek preda mnom nije jedan pravi rasni *Mek*, već običan PC klon.

IME MAJKE – LISA

Kada bi kompjuteri imali rodni list, *Macintosh* bi u rubrikama ime oca i majke imao upisano: *Xerox PARC* (Palo Alto Research Center) i *Apple Lisa*. Naime, gotovo sve novo što je *Mek* ponudio, izmišljeno je u Xerox-ovim laboratorijama, a prvi put spakovano u celinu koju zovemo računar u Apple-u i nazvano Lisa.

Najvažnija novost koju je Lisa donela sa sobom danas predstavlja našu kompjutersku svakidašnjicu. Grafički korisnički interfejs, ili još bolje rečeno operativni sistem zasnovan na komunikaciji sa korisnikom preko prozora, ikonica i menija do tada nisu bili viđeni. Svi komentatori pisali su kako se radi o idealnom načinu da se u kancelarije masovno uvedu kompjuteri. Na žalost, 1983. godina kada se kratko vreme *Apple Lisa* prodavala, cena od 10 hiljada dolara bila je sve samo ne mala.

Godinu dana kasnije, u sam osvit 1984. rodio se Apple Macintosh. Za razliku od glomazne Lise, računar je bio malen, radio je brže i, što je najvažnije, koštao je samo 2.495 dolara. Od „majke" je nasledio puno toga dobrog, uključujući grafički korisnički interfejs i miša.

Ako uporedimo „meka" sa drugim ličnim računarima toga vremena, primećujemo da je u nekim stvarima samo za njansu slabiji, ali zato u drugim bitno prednjači. U stvari, glavni „štos" konstruktora „meka" je skriven u ROM-u. Dok je tada PC XT imao ROM od 48 KB i u njemu uglavnom bejzik, u 64 kilobajte „mekovog" ROM-a bio je zapisan deo operativnog sistema bez koga nije bilo moguće realizovati grafički korisnički interfejs.

Pokušaću da budem malo precizniji. U ROM-u je upisan niz rutina jednostavno nazvanih *Mac Toolbox* i sortiranih u nekoliko grupa – *Quick Draw* (zadužena za grafiku), *Device Manager* (komunikacija sa hardverom na visokom nivou), *Window Manager*, *Menu Manager* i *Font Manager*. Za poslednja tri sami pogodite čemu služe.

Da li vas ovakva organizacija podseća na nešto? Svakako, *Mac Toolbox* je prvi API na svetu. Eto, do pre izvesnog vremena kada sam počeo da skupljam materijal za ovaj tekst ni na kraj pameti nije mi palo da je prvi API napisan mnogo pre no što se neko setio da ga tako „krsti".

ŠTA JE MEK PRVI DONEO

Macintosh je iz korena promenio filozofiju rada za računarom i prilagodio je korisnicima koji relativno malo znaju o hardveru i softveru. Danas teško da bi mogli da zamislimo život bez sledećih inovacija:

- *Cut-and-paste*
- *Undo*
- *Point-and-shot* način izbora iz menija
- Ugrađena podrška za video
- Kontrola glasom
- API
- Ugrađena podrška za rad u mreži
- Ugrađen SCSI
- *Plug-and-play* filozofija dizajna
- Dinamička alokacija memorije
- Hipertekst
- Rad sa 24-bitnom paletom boja

ISPLETENA MREŽA

U januaru 1985. godine Apple je predstavio Macintosh Office, paket u kome se nalazila jeftina lokalna mreža i laserski štampač. Doduše, skoro sedam hiljada dolara vredeo je u to vreme LaserWriter, ali to je bila prava bagatela u odnosu na druge slične mašine. Isto tako mreža *AppleTalk*, kasnije preimenovana u *Local Talk*, pružala je mogućnost lakog povezivanja računara i solidne performanse.

Za mene kao tipičnog predstavnika PC sveta, zadivljujuća je osobina "mekova" da se bez po muke umrežavaju. U samu osnovu računara ugrađena je ideja koju bi najlakše mogli prevesti: „sve što vam može zatrebati već je u



Makovi koji su otvorili vrata velikog tržišta: modeli II i SE



Sa modelom Quadra uveden je procesor Motorola 68040



Jedan iz plejade jeftinih modela: Macintosh IIsi

kutiji" Za povezivanje Macintosh računara dovoljno je sa zadnje strane pronaći odgovarajuću utičnicu „ubosti" kabl i – to je sve Od sedme revizije operativni sistem u sebi ima *peer to-peer* mrežu

Nekako istovremeno na tržištu se pojavio *Aldus PageMaker* program koji je „meka" učinio slavnim Po prvi put se na malom računaru mogla „složiti" stranica koja izgleda kao da je izašla iz nekog magazina Kada se ova vremenska distance pogledamo sasvim je jasno da stono izdavaštvo nije ni moglo da se pojavi na nekoj drugoj platformi

PIONIRSKI ZADATAK

Za ovih deset godine *Macintosh* nije prestajao da bude rodonačelnik u razvoju personalnih računara Mnoge stvari koja plejada vrhunskih PC kompjutera danas ima prvo su se pojavile upravo ovde Tako je 1986 godine predstavljen *Macintosh Plus* sa ugrađenim SCSI vaznikom za periferna uređaje

Godinu dana kasnije na tržištu su sa pojavili *Macintosh II* i *Macintosh SE* Kapacitet ROM a je povećan na 256 kB dodate zu bolje mogućnosti upravljanja tekstom filozofija lakog povezivanja još više je unapređena a na monitoru su se pojavile boja

Novim „mekovima" najviše su se obradovali umetnici Obrada slika i teksta postala je još lakša, a industrija softvera i hardvera znala je da novi polet korisnika i te kako iskoristi Tako je QMS predstavio *ColorScript 100* štampač u boji Posle njega svi su brže-bolje pojurili da izbace svoje verzije printera koja su omogućavale da se šarena sličice sa monitora prenesu na papir

Naravno, paleta od 256 boja brzo je postala tesna za sva ona koji su papir zamenili kompjuterom u svom dizajnerskom poslu Tako je 1989 godine Apple predstvio *32-bit Quick Draw*, fajl koji nadograđuje standardni *Quick Draw* do dvadesetčetvorobitna palete I naravno, opet na snagu stupa lakoća upravljanja ovim računarom Na primer ako radite na retuširanju fotografije u boji na raspolaganju vam je 24-bitna palata ali ako posle toga pređete u program koji pruža zadovoljstvo gledanja video klipova automatski se paleta sužava na zadovoljavajućih 18 bita

PONESITE GA SA SOBOM

Macintosh nije doneo samo revoluciju među stone lična računara Svog udela ima i u razvoju prenosnika Prvi takav računar prikazan ja septembra 1989 godine Mac Portable je bio „prenosnik" samo po imenu i izgledu Težina od oko sedam kilograma nikako nije bila prijatna za putovanja I pored fijaska na tržištu, to je bila dobra lekcija za kompaniju

Dve godine potom pojavio se *notebook* model težak manje od tri kilograma U atraktivno dizajniranom kućištu sa kuglicom koja zamenjuje miša, mesto je našao Motorolin 68030 procesor faks-modem I softver za povezivanje sa drugim računarom

HRONOLOGIJA

1984 Prvi Macintosh doneo je sa sobom grafički korisnički interfajs i miša u svet personalnih računara Procesor je bio Motorola 68000 sa radnim taktom 7,83 MHz, flopi disk primao je 400 kB podataka, u ROM-u se našlo mesta za 64 kB koda a kapacitet RAM a je 128 kB Cena – prihvatljivih 2 495 dolara

1987 Macintosh II dobio je SCSI hard disk, procesor 68020 na 16 MHz i matematički koprocesor 68881 Operativna memorija mogla se na dograditi do 128 MB Na matičnoj ploči mesta se našlo i za šest NuBus slotova Za ovakav računar trebalo je platiti 5 498 dolara Iste godine prikazan ja i Mac SE – klasično kućište ja redizajnirano a u njemu su se našla sva poboljšanja iz modela Mac II

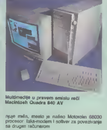
1988 Macintosh IIx dobio je još jači procesor Motorola 68030 Onaj ko je te godine bio spraman da potroši 7 769 dolara na ovaj računar pored već postojećih mogućnosti dobio je i novi flopi-disk kapaciteta 1 44 MB

1989 Mac SE/30 bio je prvi model visokih performansi koji se prodavao u tzv kozvanom „kompaktnom" kućištu Hardvar je bio gotovo identičan modelu IIx s tim što je dodate posebna video memorije Macintosh IIci doneo je nova ubrzanje (radni takt „podignut" je na 25 MHz) i bolji video podsistem ROM je proširen na 512 kB Cena za ovakvu „alu" je bila 6 969 dolara

1990 Filozofija kompanije se menja i počinje prodaja jeftinih Mekova Model Classic mogao se kupiti za samo 999 dolara Ta godine pojavili su se još i modeli LC i IIsi

1991 Napokon se pojavila mašina se 68040 procesorom i novim imenom – Quadra ROM je ponovo proširen ovog puta na 1 MB rad sa 24-bitnom paletom boja je postao standard a opcija mrežnog povezivanje proširena je opcijom za Ethernet. RAM je porastao do standardnih 4 MB a hard diskovi do 80 I više megabajta Cena od 1 269 do 8 499 dolara.

1993: Macintosh Quadra 840 AV donela je multimediju u punom svetlu Da bi sve radilo kako treba, ROM ja proširen na 2 MB procesor je 68040 na 40 MHz, operativna memorija je kapaciteta 8 MB a hard disk može da primi 230 MB podataka i programskog koda U mašinu je ugrađen hardver za rad sa video signalom, a Digital Signal Processor omogućava prepoznavanje glasa Svemu još dodajte CD-ROM faks i modem Cena ova mašine je 4 069 dolara.



Multimedije u pravom smislu reči Macintosh Quadra 840 AV

SYSTEM 7

Zajedno sa razvojem samog računara, rastao je i operativni sistem Poslednje verzija nazvana *System 7* promovisana je maja 1988 godine a do korisnika je stigla tri godine kasnije posle nekoliko odlaganja (Zlobnici bi rekli da Bil Gejts nije jedini koji obećava a zatim kasni sa isporukom) To je tridesetdvobitni operativni sistem i sobom je doneo *drag-and-drop* štampanje novi SCSI drajver, novi skup grafičkih rutina i ugrađeni „kooperativni" višeprogramski rad

Naravno, nije se tu stalo Trenutno je aktuelna revizija 7 1 koja je unapredila rad sa fontovima i promovisala *System Enabler* – malu binarnu datoteku koja sadrži hardverski zavistan kod

KUDA DALJE?

Za budućnost Macintosh-a na treba brinuti Već danas su najjači modeli (*Quadra 840 AV* i *Quadra 900 AV*) potpuno opremljeni za multimediju naročito u pogledu obrade televizijskog signala. Naravno primenjen je već dobro oprobani recept sva što vam je potrebno već je ugrađeno

Evo, već je treća godina kako Apple zajedno sa kompanijama IBM i Motorola radi na razvoju nova generacije „mekova" koji će koristiti PowerPC procesore O njima ste već dosta u „Računarima" čitali i verujem da su vam sve informacija o tome poznata Zato evo još nekoliko reči o nečem sasvim novom *Newton Technology* je novi koncept operativnog sistema u čijem središtu nije dokument nego informacija. Za sada je javnosti dostupna verzija namenjena malim personalnim elektronskim rokovnicima ali će vremenom biti implementiran i na veće modele poput Macintosh-a.

I na kraju moram priznati da i posle deset godina potajno žudim za jednim Macintosh-em na stolu Još kada bi bio povezan u mrežu se dobrim laserskim štampačem potpuno opremljen za multimediju mojoj sreći kraja ne bi bilo Na žalost kako stvari stoje, ostaće to samo sanak pusti Preostaje mi samo da pošaljem rođendansku čestitku „Mek želim ti da još dugo godina donosiš nove tehnologije u svet ličnih računara"


NA PRAVOM M
GROUP
Beograd, 29. Novembra 43/1 tel: 322-19-59; 322-


MESTU

"Rat" tekst procesora nikada nije bio tako žestok kao poslednjih meseci – izgleda da svi konkurenti očekuju da upravo ovih dana tržište izabere program koji će tokom sledećih godina dominirati *Windows* scenom, pa se svako bori za što bolju startnu poziciju. *WordPerfect 6.0* za *Windows*, uzdanica korporacije *WordPerfect*, svakako je najveći i najmoćniji, ali i najglomazniji tekst-procesor sa kojim smo se do sada sreli.

Originalni pristup dizajnu tekst procesora kao da je izgubio na značaju – svi konkurenti su se opredelili za relativno sličnu strategiju tj. za pisanje „gigantskog" programa u koji će biti ugrađena baš sve što bi nekome ko piše i prelama tekst ikada moglo zatrebati. Na konzumirane resurse se uopšte ne gleda, a prema onima čiji hardver makar za korak zaostaje od nekog svetskog poslovnog standarda kao da više nema milosti. *WordPerfect* 6.0 za *Windows* zahteva brz 80486 računar sa 8 megabajta memorije i diskom od dvestotinak ili više megabajta!

BEZ ALATA...

Ove hardverske zahteve nećete međutim pročitati na pakovanju novog *WordPerfect*-a. Tamo se pominje 80386 računar sa 4 megabajta memorije i trideset slobodnih megabajta na disku. Lepo zvuči ali predstavlja tek marketinški slogan: novi WP 6.0 će istina raditi i na toj konfiguraciji, ali tako sporo da o tome jedva vredi govoriti. Ne radi se o nešto sporijem izvođenju specifičnih operacija nego o usporenju koje čini bilo kakav rad nemogućim: ispisivanje menija, umetanje slova pa i samo pomeranje kursora upaliće lampicu na disku i neće je gasiti više sekundi. Pri eksperimentu na 80386/40 računaru sa 4 megabajta memorije desilo nam se da učitavanje sasvim kratkog teksta potraje preko tri minuta! Ako je konfiguracija spora, a ipak želite da koristite novi *WordPerfect*, svakako učinite što možete po pitanju memorije: dodatna četiri megabajta obično znače više od sledeće generacije mikroprocesora, tako da je rad na 386/40 sa 8 megabajta RAM-a kako-tako moguć, ali i dalje daleko od komfora – tek će 80486 računar na 50 ili 66 MHz učiniti da tekst „leti" kao što je leteo nekada u *WordPerfect*-u 5.1 za DOS.

WordPerfect je dakle po hardverskim zahtevima iznad *Microsoft*-ovog *WinWord*-a 6.0 koji formalno radi i na 80286 računaru, ali su u osnovi hardverski apetiti slični: oba giganta traže mikroprocesor najnovije generacije i dovoljno radne memorije. Uz to dobro dođe i neka brža grafička kartica, mada moramo da kažemo da smo imali raznih problema sa *WordPerfect*-om 6.0 i drajverima za razne grafičke kartice – najviše su nas namučile Trident-ove kartice, dok je najbolje išlo sa „anl čkim". Tsaug ovime u rezoluciji 800×600. Problemi su obično rešavani nabavkom nove verzije drajvera i njenom ugradnjom u konfiguraciju *Windows*-a.

Dejan Ristanović

Pošto su hardverske brige prebrinute, novi *WordPerfect* treba instalirati, što se pokazuje jednostavnim ali prilično dosadnim poslom: procedura je automatizovana, ali potraje preko 40 minuta! Optimista ste ako mislite da je *Windows* višeprogramski operativni sistem i da biste, dok se sadržaj disketa kopira na disk, možda mogli da radite i nešto drugo: instalaciona procedura tako „zatrpa" *Windows* da je čak i pasijans na brzom 486 računaru neshvatljivo spor (na *Pentium*-u bi možda išlo brže, ali na žalost nismo imali priliku da probamo). Kada li će konačno *Windows* multitasking prestati da se „guši" čim jedan od procesa pristupi disk jedinici ili serijskom portu?

Ako se opredelite za automatsku i punu instalaciju, ostaćete bez oko 33 megabajta prostora na disku – kada samo pomislimo kako su mnogi nekada bili srećni sa diskom od 10 megabajta... Sa *Custom* instalacijom možete da prođete znatno „jeftinije" – pruža vam se mogućnost da izaberete koje ćete komponente paketa instalirati, pri čemu zauzeti prostor varira između 10 i 33 megabajta. Pokazuje se na žalost da je većinu komponenti šteta preskočiti – glavni kvalitet novog *WordPerfect*-a je baš u tome što nije puki program za pisanje, crtanje, štampanje nalepnica... Eventualno se možete odreći tutorijala, slika i primera i tako svesti instalaciju na oko 25 megabajta, što je otprilike optimalna ravnoteža između resursa i mogućnosti.

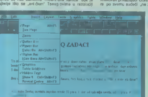
Zanimljiva novost je instalacija na više diskova – ako na jednoj particiji nemate dovoljno prostora za čitav *WordPerfect*, njegove komponente možete da rasporedite na drugim diskovima ili drugim particijama, pri čemu se na ekranu stalno ispisuje koliko gde ima slobodnog prostora. Pošto se opredelite za raspored koji najbolje „pasuje" vašem sistemu, kreće umetanje disketa i 40 minuta instalacije. Istom procedurom kasnije možete dodatno instalirati neke druge komponente paketa, ali je pri tome primenjen metod koji smo kritikovali prikazujući OS/2 2.0: instalaciona procedura po svemu sudeći „ne zna" na kojoj je disketi šta, pa će vas usterati da umećete svih dvanaest 1.44 M disketa – mnoge od njih će se samo za trenutak „u prazno" zavrteti uz zahtev da se ide dalje.

Najbolje što možete uraditi kada završite instalaciju jeste da startujete program, malo prošetate po menijima i onda ugasite računar i „zagrejete" stolicu: ovaj paket se naprosto ne može upoznati bez uputstva za upotrebu! Lutajući po menijima ćete prirodno naučiti da obavljate najosnovnije stvari, ali da ste želeli osnovne stvari sasvim bi vam lepo poslužio i *WordPerfect 5.1* za DOS! Glavne prednosti verzije 6.0 sastoje se u integraciji pojedinih komponenti, tako da se ponekad tek ispod druge ili treće „generacije" menija krije neka važna mogućnost ili eksterni program. Ukratko, pročitajte *WorkBook*, isprobajte prve lekcije *Coach*-a i tek onda krenite na posao, držeći *Reference manual* nadohvat ruke.

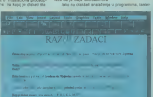
OSNOVE WORDPERFECT-a

Glavni ekran *WordPerfect*-a 6.0 za *Windows* ne razlikuje se previše od ekrana bilo koje standardne aplikacije – meniji u vrhu, tasterski meni ispod njih, veći deo ekrana posvećen tekstu koji se obrađuje, statusna linija u dnu i trake za skrolovanje desno od nosno dole. Novitet verzije 6.0 je takozvani *Power Bar*, neka vrsta tasterskog menija druge generacije. O čemu se radi? Programima se nekada upravljalo komandama kao što su Ctrl J, Ctrl K i tome slično. Kako su programi rasli tako je i funkcija bilo sve više pa ih je malo ko mogao sve popamtiti – pojavili su se „padajući" meniji u okviru kojih su funkcije zamenjene odgovarajućim rečima. No, programi su i dalje rasli pa su i meniji postali preveliki i nepregledni – početniku nije bilo lako da pogodi u kom je meniju funkcija koja mu je potrebna i kako se tačno zove. Zato su prvi *Windows* tekst procesori, posebno *WinWord* i *WordPerfect for Windows* uveli tasterske menije: pri vrhu ekrana se pojavljuje nekoliko „tastera" sa jasnim vinjetama i natpisima preko kojih se pozivaju najosnovnije funkcije tekst procesora, samo će oni kojima je potrebna neka specifičnija usluga morati da se „obrate" meniju. Nije nam poznato ko se prvi dosetio tasterskih menija, ali mnoge indicije govore u prilog teze da je ideja *Microsoft*-ove.

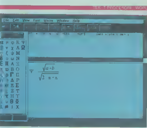
Iako su olakšali snalaženje u programima, taster-



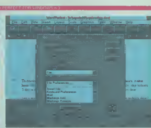
Savršeni dizajn. Radni ekran WordPerfect-a



Pravi WYSIWYG na ekranu se vide razne naglašavanja, ali ćirilična slova i dalje ne izgledaju savršeno



Šarenije, ali bez bitnijih noviteta: editor formule



Puna konfigurabilnost: editovanje tasterskih menija

ski meniji su tokom poslednjih godina pokazali i određene slabosti, najveća slabost je pomalo paradoksalno što su se korisnicima svideli i što se kod njih stvorio utisak da u tekst procesoru i nema ničega osim onoga što je dostupno preko tasterskog menija. Zato ih je trebalo u najmanju ruku duplirati: što smo primetili još kod programa *Ventura Publisher for Windows* – u gornjem redu je osnovni tasterski meni iz koga se pozivaju najvažnije komponente paketa, a ispod njega mali tasterski meni sa većim brojem sitnijih tastera preko kojih se pozivaju neke specifičnije funkcije. Slična koncepcija usvojena je i kod novog *WordPerfect*-a: glavni meniji su izgubili na značaju, ali je zato prozor za tekst dalje umanjen... prosto se pitamo da li će jednoga dana kontrolna struktura „pojesti" čitav ekran i da li će proizvođači tekst procesora od nas zatražiti da kupimo sistem sa dva monitora, pri čemu bismo na jednom gledali tekst a na drugom razne tasterske i druge menije.

Bilo kako bilo, glavni tasterski meni *WordPerfect*-a obezbeđuje uvlačenje teksta (*Indent*), ubacivanje najrazličitijih znakova za nabrajanje (*Bullet*), pristup kalendaru, štampanje koverata i cirkularnih pisama (*Envelope* i *Merge*), pozivanje programa za crtanje (*Draw Chart*), umetanje slike i okvira sa tekstom (*Figure, Text Box*) i rad sa stilovima. Ispod njega je mali tasterski meni koji obezbeđuje premeštanje i dupliciranje teksta, izbor fonta, veličine i naglašavanja slova, uravnavanje linije, razne vidove pregleda dokumenta, štampanje i mnoge druge usluge. Taj mali tasterski meni je, rekli bismo, veoma dobro osmišljen i realizovan, dok su funkcije osnovnog tasterskog menija pomalo čudno izabrane: ne možemo, recimo, zamisliti zašto je u njemu rezervisano mesto za funkcije kao što je ubacivanje datuma u tekst ili štampanje koverta, dok su daleko potrebnije komande za izbor fonta ili veličina slova „stanule" u mali tasterski meni. Ili možda da pokušamo iz čisto marketinških razloga: pošto će svaki eventualni kupac dobro znati da tekst procesori ovako ili onako omogućavaju da se izadare font ili odštampa tekst, a jasnom vinjetom sa natpisom *Envelope* mu se odmah stavlje do znanja da *WordPerfect* obezbeđuje rad sa kovertima, što je u praksi često potrebno a uvodi se tek u novijim generacijama tekst procesora. Bilo kako bilo, ne verujemo da osnovni tasterski meni treba „ostaviti na miru" – u njega se ubacuju važnije stvari. Srećom, u *WordPerfect*-u je to ubacivanje sasvim pojednostavljeno!

WordPerfect je zapravo konfigurabilan u meri koja je do skora smatrana nemogućom: možete maltene sve promeniti i prilagoditi svojim potrebama! Čak ni osnovni meniji nisu fiksirani... mogu im se dodati razne usluge, premestiti neke stavke u druge menije pa čak i kreirati sasvim novi meni. Pokušali smo da prevodimo neke od menija na naš jezik i pri tome su se pojavili razni tehnički problemi, ali smo utiska da bi se taj posao mogao uspešno obaviti već i na osnovu dokumentacije koja stiže uz paket – kod prethodnih verzija bila je neophodna nabavka posebne tehničke dokumentacije za one koji se bave razvojem programa koji „prate" *WordPerfect*.

Kada se radi o tasterskim menijima, možete ih napraviti više (zapravo, uz sam paket dobijate nekoliko) i u svaki od njih umetati ne samo funkcije samog paketa nego i svoje makroe i druge *Windows* programe. Izabrane funkcije možete slobodno rasporediti, a novitet verzije 6.0 je mogućnost da između dva tastera odgovarajućeg menija ugradite širu ili užu „rupu" koja će učiniti sam meni grafički dopadljivijim i smanjiti verovatnoću da mišem „pogodite" susedni taster. Same tasterske menije možete da raspoređujete po ekranu – umesto uravnate linije možete ih grupisati u prozore, „smaknute" trake i mnoge druge strukture. Konfigurabilnost ponegde dostiže čak i pomalo parverzne razmere: možete da izaberete font i veličinu slova kojom se ispisuju razne komponente običnih i tasterskih menija! Probali smo, na primer, da sve ispise prebacimo u ćirilično TTF font i *WordPerfect* je za čas prestao da liči na sebe.

Moramo priznati da smo, prelistavajući razne opcije *WordPerfect*-a, sa žalnjom tražili poglavlje koje bi se odnosilo na boje. Jer, *WordPerfect* u tom domenu već duže vreme ne odgovara našem ukusu: stari dobri *WordPerfect 5.1* za DOS je omogućavao da podesimo boje svih elemenata ispisa. U verziji 5.1 za *Windows* „izgubilo" se ono najvažnije – mogućnost da se definiše boja osnovnog teksta i njegove pozadine: pozadina je uvek morala da bude bela, a tekst crn. Iako se mnogima ova kombinacija, koja podseća na beli papir i crnu olovku, izrazito dopada, nas je oduvek iritirala: mnogo smo lakše radili sa svetlim tekstom na tamnoj pozadini! Onda je došao *WordPerfect 6.0* za DOS koji je obezbedio konfigurisanje svih mogućih boja kojima se ispisuju meniji i njihovi najsitniji delovi, ali se bojom osnovnog teksta i dalje nije moglo upravljati! Prosto neverovatno! U *Windows* verziji stvari su srećom došle na svoje mesto – treba aktivirati stavku *Preferences* iz **File** menija, kliknuti na *Display* i zahtevati ispisivanje u *Windows* sistemskim bojama. Kasnije u *Control Panel*-u *Windows*-a podesite željene boje ili izaberete neku od gotovih paleta i *WordPerfect* to poštuje – začas smo dobili željena plava slova na crnom ekranu uz osenčene menije. Trebalo je da *WordPerfect* to obezbedi još odavno, ali... bolje ikad nego nikad.

NAJVAŽNIJI NOVITETI

Najvažniji noviteti *WordPerfect*-a 6.0 za *Windows* svakako su ugrađeni elementi iz paketa *PlanPerfect*, *DrawPerfect* i *WordPerfect Presentation*. *PlanPerfect* je „skriven" iza opcija za rad sa tabelama – umesto prostog ispisivanja linija i sadržaja polja, novi *WordPerfect* obuhvata i čitav mali *Excel*, odnosno mogućnost rekalkulacije uz unošenje razmerno složenih formula (u sistem je ugrađeno preko 100 osnovnih operacija koje se mogu kombinovati) i makroa u tabele. Tabela se više ne mora prikazati kao niz natpisa i brojeva: jednostavnim klikom na odgovarajuću opciju taj ispis se pretvara u grafik, „pitu", histogram ili neku drugu strukturu. Takvu strukturu dalje obrađujemo u *DrawPerfect*-u koji se krije iza „tastera" *Draw* – to je zgodno mesto za dodavanje natpisa, legendi, kombinovanje sa drugim slikama i vinjetama... rezultujući crtež ugrađuje se u dokument i u svakom je trenutku spreman za izmene koje će se dogoditi potpuno automatski čim se promeni neki od podataka u originalnoj tabeli.

U većini reklama pročitaćete da je *WordPerfect 6.0* opremljen brojnim OLE alatkama. Skraćenica OLE označava *Object Linking and Embedding* i odnosi se na vezu sa drugim aplikacijama. Ideja je u tome da, recimo, kreiramo neku tabelu u *Excel*-u i onda je ugradimo u neki *WordPerfect* dokument, što je dakako bilo „oduvek" moguće. Međutim, ako je tabela povezana (*linking*) sa *WordPerfect*-om, promena nekog njenog polja u *Excel*-u trebalo bi automatski da unese promenu u dokument bez potrebe da tabelu ponovo učitavamo. Korak dalje je *embedding*: kada je tabela integralni deo *WordPerfect* dokumenta ali je i dalje vezana za *Excel*, pa bi *Excel* bio u stanju da je tamo „nađe" i na odgovarajući način promeni. U teoriji ovo deluje jako lepo. U praksi... na radi. Veze su racionalne tek ako je programe pisala ista firma, a i tada ne funkcionišu baš sjajno – na raspolaganju nam je bila verzija *DrawPerfect*-a iz 1992. godine i nije bilo šanse da se slika poveže sa *WordPerfect*-om tako da dvostruki klik na nju startuje *DrawPerfect* i da eventualne izmene budu automatski registrovane u dokumentu. Možda neke od sledećih godina... Ali na nivou samog *WordPerfect*-a 6.0 OLE funkcioniše veoma dobro: program „zna" kako je koja slika urađena i automatski će pozvati sve komponente potrebne za njeno podešavanje. Što je posebno lepo, čitav posao se obavi brzo: imamo utisak znatno brže nego u *WinWord*-u, pa je i komfor pri radu veoma visok.

Interni *DrawPerfect* se „ponovio" raznim opcijama za rad sa fontovima... priloženi *TrueType* i *Bitstream* fontovi (preko 25 familija slova) obezbeđuju istezanje i deformisanje slova, definisanje obvojnica, ispis u svim pravcima i smerovima i uopšte mnoge funkcije za koje je do skora bio potreban *CorelDraw*. Stekli smo utisak da se većinom ovih usluga upravlja znatno jednostavnije nego kod *Corel*-a – mogućnosti su, naravno, znatno manje, ali su dobro izabrane pa se čak i veoma efektne slike mogu uraditi bez napuštanja tekst procesora... a može se i uštedeti i desetak megabajta koje zauzima ozbiljnija instalacija *Corel*-a!

Uz pakete za crtanje, prezentacije i unakrsne proračune, *WordPerfect* bi verovatno trebao da uključi i neku formu baze podataka, ali je to, izgleda, ostavljeno za neku od sledećih verzija. Imena i adrese se doduše mogu ugraditi u tekst i onda koristiti za štampanje cirkularnih pisama koja su oduvek bila jaka strana *WordPerfect*-a, ali to naprosto nije zamena za pravu bazu podataka. Uz *WordPerfect* ćete, dakle, na disku morati da držite i *MS Access* ili neku drugu *Windows* bazu podataka. *WordPerfect* na žalost ne obezbeđuje direktan „uvoz" datoteka u *Access*, *dBASE* ili bilo kom drugom formatu – u okviru same baze podataka mora se formirati upotrebljiv tekst fajl.

GOTOVI UZORCI

Važan novitet *WordPerfect*-a 6.0 su uzorci za tekstove, takozvani *templates*. Moramo da se pohvalimo da smo nešto slično predložili još pre šest godina, u

knjizi „Obrada teksta na računaru" (izdanje „Tehničke knjige" davno rasprodato), tada je bio aktuelan *WordPerfect 4.2* pa smo čitaocima preporučili da pripreme zaglavlja za nekoliko tipova dokumenata: poslovno pismo, zapisnik, memorandum, izveštaj... kada kasnije treba kucati neki od tih dokumenata, učita se „matrica", snimi pod novim imenom i odmah krene sa radom! *WordPerfect* je došao do slične ideje ali je implementacija daleko kompleksija – kreiraju se posebni stilovi za svaki tip teksta, i onda se pri kreiranju dokumenta samo specificira željeni *template* a sam *WordPerfect* obavi čitavo prilagođenje izabranom formatu. To je izgleda *WordPerfect*-ov odgovor na *Microsoft*-ove *Wizard* programe ugrađene u *WinWord 6.0*.

Iako su osnovne usluge tekst procesora oduvek bile na nivou, *WordPerfect 6.0* za *Windows* tu donosi nekoliko zanimljivih unapređenja. Jedno od njih je detaljna statistika teksta – ne radi se samo o broju reči i slova, nego i o strukturi pasusa, rečenica i raznim drugim statističkim parametrima. Analizom statističkih parametara teksta autor može da uoči neka od svojih grešaka (npr. predugačke rečenice, pasuse koji se sastoje od premalo ili previše teksta i tome slično) i tako radi na poboljšavanju svoga stila. Da bi statistike bile tačna važno je da *WordPerfect* „zna" šta je slovo, a šta ne, što znači da ne YUSCII treba zaboraviti, o prilagođenju *WordPerfect*-a 6.0 domaćim uslovima govorimo u posebnom prilogu.

Oni koji pišu na engleskom i koje YUSCII ne brine imaju dodatne mogućnosti za kontrolu teksta i stila. To su pre svega program za kontrolu spelovanja i rečnik sinonima, a zatim i paket *Grammatik 5*, program za kontrolu gramatike i stila teksta pisanog na engleskom jeziku. Program *Grammatik* je *WordPerfect* nedavno licencirao od firme *Reference Software*, a zatim mu se paket toliko svideo da je kupio čitavu firmu *Reference Software*. Rezultati se primećuju – za razliku od verzije 6.0 za DOS kod koje je *Grammatik* samo „strano telo" prikrpljeno programu koje se u svakoj komponenti dizajna razlikuje od njega, novi *Grammatik* je solidno integrisan ne samo u *Windows* nego i u *WordPerfect* – prava *Windows* aplikacija koja radi u svom prozoru ali može direktno menjati osnovni tekst koji se obrađuje. Teško je reći ima li suštinskih izmena u programu *Grammatik* – iako je grafički izgled promenjen a upotrebe olakšane, imamo utisak da je radni deo ostao isti. I dalje se tekst pregleda, greške prijavljuju i (automatski ili ručno) ispravljaju, a korisniku je stavljena na raspolaganje i mogućnost da ukine pojedine poruke i tako prilagodi *Grammatik* svom stilu pisanja – ne mogu se, na kraju krajeva, očekivati iste primedbe na neki tehnički tekst, poslovno pismo i roman!

Kada već govorimo o dokumentaciji, da pomenemo novitet na koji je potrošeno dosta prostora: de... pisanje „pozadine" teksta tj. slike, vinjete ili natpise koji će se štampati „ispod" osnovnog dokumenta stvarajući tako utisak štampe na nekom memorandumu ili posebnom papiru. Naravno da za punu ubedljivost ovih oznaka (nazvanih *Watermarks*) treba posedovati kvalitetan laserski štampač – na nekom devetopincu „vodeni žig" liči na običnu mrlju.

KIŠTA NOVO KOD FORMULA...

Kada se radi o formulama, novi *WordPerfect* predstavlja svojevrsno razočaranje. Pokazuje se zapravo da su formule oduvek „pastorče" *WordPerfect*-a: najpre smo čekali sve do verzije 5.1 da editor za formule uopšte bude ugrađen, a onda se u novim verzijama taj editor jedva primetno unapređuje. Formule su i dalje samo slike posebnog tipa, a njihov unos i ispravljanje i dalje zahteva solidno matematičko znanje: kuca se neka vrsta „bejzik" ekvivalenta, pa bi a+b bio običan zbir, sqrt a+b uvodi simbol za kvadratni koren, sqrt sqrt(2) over sqrt 3 bi bio dvostruki razlomak pod korenom i tome slično. Za početak treba zapamtiti da se umesto običnih izraza uokviravaju vitičastim zagradama { } i da se tek komandom dobija izgled trenutno unesene formule uz naravno moguću poruku o sintaksnoj grešci *Incorrect format* – ni tekst procesor više ne može da prođe bez provere sintakse.

WordPerfect 6.0 je solidno dopunio arsenal simbola kojima su raspolagale prethodne verzije: uz tradicionalna grčka slova i matematičke operatore tu su sada oznake raznih funkcija, specijalni znaci, višestruke oznake za integrale i sume i tome slično. Značajno je što se u formule sada mogu ubacivati i svi *WordPerfect* karakteri, što otvara mogućnost editovanja formule u kojima figuriše ćirilični tekst, slova čćđšž ili čak neki japanski ili kineski znaci. Preciznije se upravlja osnovnim fontom formule i razlikama u veličinama „susednih" simbola. Pa ipak, korisnik i dalje ne može da definiše veličinu svakog pojedinačnog znaka, što neretko smeta kod razlomaka čiji je, recimo, brojilac jako jednostavan a imenilac je neki složen višestruki razlomak sa korenima. Kada bismo to pisali rukom, brojilac bismo verovatno napisali „većim fontom" kako bi čitav razlomak bio simetričan.

Početnike zbunjuje pomalo nepotrebna „simetrija" između formule i ostalih ilustracija – i formule se mogu privezati za stranu (što ni za šta ne služi) para... grafički znak. Ukoliko se formule mešaju sa tekstom, *WordPerfect* će automatski razmicati redove, što često ne obezbeđuje baš najdopadljivije dokumente. Zato cete obično koristiti formule vezane za pasus.

WORDPERFECT I YU SLOVA

Počevši od verzije 5.0, *WordPerfect* u potpunosti podržava pisanje tekstova na našem jeziku – uz latinična slova čćđšž, u set znakova je ugrađena i potpuna ćirilica, kako ruska tako i srpska, makedonska, bugarska... samo zadamo kod slova i ono se pojavi u tekstu!

Sve ovo je sa koncepcijske strane lepo zamišljeno, ali nikada nije baš sjajno funkcionisalo: ćirilična slova se nisu videla na ekranu, na štampaču su znatno odudarala od okolnog teksta i tome slično, bilo je potrebno uložiti dosta rada da *WordPerfect* "progovori srpski". U verzijama 5.1 i 5.2 za *Windows* i verziji 6.0 za DOS unesena su razna poboljšanja koja pre svega počivaju na činjenici da *WordPerfect* radi u grafičkom modu i da može iscrtati bilo koje slovo. U ovoj *Windows* verziji stvari su dovedene gotovo do savršenstva, pre svega zahvaljujući *Bitstream* fontovima koji se isporučuju uz paket.

Sve što treba uraditi jeste priprema drajvera za tastaturu: aktivirajte opciju *Preferences* iz **File** menija, kliknite na *Keyboard*, napravite novu definiciju tastature i onda je "punite" slovo po slovo. Sve pomalo liči na poznati *ParaWin*: omogućite predefinisanje tastera sa slovima, kliknete na odgovarajuće slovo, pređete u *Keyboard script* prozor i jemo pritiskom na Ctrl W ubacite odgovarajući kod. Kodovi latiničnih slova čćđšžČĆĐŠŽ su respektivno 1 99, 1 97, 1 79, 1 177, 1 207, 1 98, 1 96, 1 78, 1 176 i 1 206, dok se ćirilična slova nalaze u *Cyrillic* tabeli. Posao je jednostavan ali prilično dugotrajan i dosadan, naročito kada treba kliknuti na 60 slova i pronaći ih u tabeli, srećom to se radi samo jedanput.

Pošto je drajver kreiran, pritisak na odgovarajući taster proizvodi željeno slovo koje će biti prikazano na radnom ekranu, tokom *preview*-a, štampano na papiru... *WordPerfect* automatski aktivira potrebne fontove tako da se bez problema obrađuje tekst koji se sastoji od znakova različitih alfabeta. Novo u verziji 6.0 je da se naša slova mogu ugraditi u matematičke formule, legende dijagrama i tome slično, a podržana su pri sortiranju, konverzijama i svim drugim operacijama.

Jedino opravdanje za konstataciju da su "stvari dovedene **gotovo** do savršenstva" je izgled ekrana: naša slova, a posebno ćirilica, izgledaju savršeno na papiru, ali na ekranu pomalo odudaraju od okolnog teksta. Glavni problem nije u obliku nego u veličini: pošto su slova iz nekog razloga nešto sitnije od okolnih – ne toliko da bi bila teško čitljiva, ali sasvim dovoljno da se primeti da tu nešto "nije u redu". Zato će neki korisnici verovatno odlučiti da koriste ćirilične i latinične *TrueType* fontove koje su nabavili nezavisno od *WordPerfect*-a, čemu se ovaj paket savršeno prilagođava. Pa ipak, prava je šteta lišiti se izuzetnih *Bitstream* fontova koje dobijate uz *WordPerfect*.

Podrška stranim slovima vidi se i kod komunikacije *WordPerfect*-a sa DOS i *Windows* okruženjem – pri snimanju i učitavanju teksta može da se izabere bilo koja od DOS i *Windows* kompatibilnih kodnih stranica, i strana slova više neće biti zamenjene ružnim kvadratićima nego smislenim slovima. Nije dokumentovan način da se uvode nove kodne stranice, što će smetati korisnicima koji su navikli na raspored YUSCII, Custom 2 Računari ili možda na naš zvanični standard JUS 015. Verujemo ipak da će se brzo naći način da se i taj nedostatak ipak nadomesti.

WordPerfect-ov editor formule bio je sjajan 1990. godine kada se pojavio, ali bi mu danas dobro došla neka unapređenja. Pre svega, bilo bi zgodno da se formula može uneti u osnovnom tekst modu, a ne u okviru posebnog ekrana. Trebalo bi, kao što smo videli, omogućiti korisniku da upravlja svakim pojedinačnim elementom formule, mada bi takvo upravljanje možda učinilo program komplikovanijim za korišćenje.

...I PREVIŠE NOVOG KOD MAKROA

Pošto smo kritikovali nedostatak promena u editoru formula, prelazimo na kritiku prevelikih promena makroa – baš je svetu teško ugoditi. Još od verzije 4.2 svaki novi *WordPerfect* dobija potpuno novi makro jezik, što se moramo reći graniči sa neozbiljnošću. Makro jezik naime nije nešto što bi se sastojalo od svega nekoliko komandi i što bi se naučilo brzim pogledom u HELP – to je čitav programski jezik za manipulaciju sa tekstom koji uključuje konstante, promenljive, procedure, funkcije, kontrolne strukture, čak i takve stvari kao što je generator slučajnih brojeva. Nije nam jasno odakle korporaciji *WordPerfect* smelost da od nas očekuje da kod svake nove verzije „zaboravimo" prethodni jezik i ponovo učimo neki novi, čak se i makro jezik ugrađen u *WordPerfect* za DOS temeljno razlikuje od svog *Windows* „konkurenta". Uz paket se doduše dobija program koji prevodi stare makroe (i to ne sve!) na novi jezik, ali nam program neće pomoći da i naše znanje *upgrade*-ujemo. Tome će poslužiti dokumentacija koju da bi stvar bila još gora treba posebno naručiti. *WordPerfect* je izgleda mišljenja da su prosečnom korisniku makroi korisni, ali da ih ne mora sam praviti – neka kupi pakete makroa od same korporacije ili nekog nezavisnog proizvođača.

Rešenje ovoga problema bilo bi u konstruisanju jednog programskog jezika opšte namene koji bi bio specijalizovan za rad sa tekstom – recimo neki *Word Basic* ili *WordPerfect Basic*. Taj bi jezik iz verzije u verziju trebalo unapređivati, ali uz zadržavanje pune vertikalne kompatibilnosti – korisnik bi uvek mogao da koristi ono što je ranije naučio, a novitete bi usvajao kada bi mu zatrebali. Imamo utisak da *Microsoft*, kada se o makroima radi, sledi koncepciju sličniju ovoj.

PAKET ZA SVE POTREBE

WordPerfect 6.0 je sjajna alatka za prelom teksta – uz dovoljno brz računar, čak i prelom složenih stranica sa više kolona i slika predstavlja prilično jednostavnu operaciju. Naročito je dobro rešen rad sa fontovima – *WordPerfect* karakteri se automatski mapiraju u potrebne *TrueType* karaktere tako da su latinični, ćirilični, grčki pa čak i arapski znaci ujednačeni po grafičkom izgledu, širinama i *kerning*-u. Sama štampa može da se poveri kako *Windows*-u, tako posebnom drajveru koji se dobija uz *WordPerfect*: bolji rezultat dobija se primenom ovog poslednjeg. Stvar je naime u tome što korporacija *WordPerfect* ima zbilja ogromno iskustvo u pisanju drajvera za štampače – zli jezici govore da *Microsoft* naprosto nije mogao da parira tom iskustvu i da je zato napravljen *Windows* koji je pitanje drajvera prebacio u nadležnost proizvođača opreme. Nevolja je u tome što su firme proizvođači hardvera nateran da pišu softver, pa je kvalitet svih drajvera znatno opao. No, *WordPerfect* se ne predaje – uz tekst procesor se sada isporučuje „konkurentski" drajver koji se koristi isključivo iz *WordPerfect*-a. Kvalitet je tako poboljšan, a prostor na disku – koga briga za to?

Završili ovaj prikaz tvrdnjom da je *WordPerfect* dobar ili odličan program za obradu teksta naprosto je deplasirano, pošto se kvalitet podrazumeva. Bolje je da završimo tvrdnjom da je korporacija *WordPerfect* složila jedan gigantski paket koji svakome korisniku garantuje da će sve njegove potrebe za obradom i prelomom teksta biti zadovoljene, ali da se ni za razne srodne i prateće usluge neće obraćati drugim programima. Mada, kada se radi o potrebama, kraj je neme... u „Računarima u vašoj kući" smo predstavljali „moćne" tekst procesore koji zamisle zauzimaju 8 ili 16 kilobajte memorije i nude „sve što je jednom piscu potrebno". Izgleda da su za ovih desetak godina i 100 brojeva našeg časopisa potrebe pisaca drastično porasle i da ih MOGU ZADOVOLJITI tek programi koji zauzimaju po 30 megabajta. Prosto ne smemo ni da pomislimo kakve će biti potrebe pisaca na ulasku u sledeći milenijum, ali se ne bismo začudili da ih ni desetak gigabajta softvera ne podmiri!

ČEKAJUĆI CHICAGO

***Windows for Workgroups 3.11* treba shvatiti kao najavu mogućnosti koje će pružiti *Chicago*, eventualno paket *Daytona*, ili *Windows NT 3.11*. Uz izostavljanje mrežnih sposobnosti i konfigurisanje za upotrebu na individualnom računaru, ovaj *Windows* je prilika da, dok čekate *Chicago*, značajno unapredite svoju kućnu mašinu.**

Zoran Kehler

Windows for Workgroups 3.1 je u *Windows* porodicu doneo integrisane sposobnosti za *peer-to peer* umrežavanje. Microsoft ga je u najavama namenio grupama sa najviše 20 korisnika koji imaju potrebu za deljenjem informacija i resursa. Za grupe veće od 20 korisnika preporučen je mnogo formalniji pristup kao što je Lan Manager, proizvod njihove kuće. Po mišljenju predstavnika Microsofta, malim grupama ni su potrebne *client server* usluge.

Pored omogućavanja *peer to peer* umrežavanja WFW je izvrsno uklopljen u Microsoftovu globalnu strategiju razvoja *Windows*-a prema *Windowsu NT* i dalje ka *Cairo* sistemu. Klijentska mašina vezana na *Windows NT* server biće za sada ili LAN Manager klijent ili će imati *Windows for Workgroups*. Ono što Microsoft ne kaže jeste da je WFW rešenje jeftinije i jednostavnije, naročito za manje zahtevne korisnike.

Najava projekta *Chicago* ili *Windows 4.0* zbunila je korisnike *Windows*-a i izdavače softvera. Isprva se činilo da *Chicago* znači paralelan razvoj dve grane *Windows* familije, a da nije jasno da li obe grane vode ka *Windows*-u NT i *Cairo* projektu. Prošlogodišnja pojava beta verzije *Chicagoa* delovala je umirujuće jer je konkretno videlo da je *Chicago* ipak prirodna evolucija *Windowsa 3.10* po arhitekturi, prvenstveno u tome što najveći deo komponenti sistema postaje 32-bitni i da je umrežavanje ugrađeno već u baznu verziju paketa.

Problemi su se pojavili kada se ispostavilo da *Chicago* neće moći da se pojavi na vreme, i da se neće uklopiti u red vožnje Microsofta za ovu godinu. Mislim da je *Windows for Workgroups 3.11* pre proizvod koji treba da popuni prazninu do pojave *Windowsa 4.0* nego projekat koji je baš sto sada bio planiran da se pojavi. Situacija potpune medijske zbrke oko novog proizvoda kao da govori u prilog ovome. Krajem prošle godine tržištem u SAD se pronele vesti da stiže *Windows 3.11*, pa su prodavci u cenovnike oglase stavili „Windows 3.11". Zatim se na Microsoftovim forumima na *CompuServe*-u i na Internetu u ftp.microsoft.com pojavila dogradnja za *Windows 3.10* u *3.11* koja, blago rečeno, nije radila ništa korisno. Na kraju, distributerima su posiale kutije u kojima se nalazio *Windows for Workgroups 3.11*.

RAZLIKE U ODNOSU NA 3.10

Verzija 3.11 proširuje 32-bitne mogućnosti *Windows for Workgroups 3.10* na podršku za 32-bitne veznike za mrežne kartice i 32-bitni pristup datotekama. Ovo su glavne novosti, ali ima i sitnijih poboljšanja sistema u oblasti umrežavanja na osnovu primedbi korisnika i proizvođača hardvera.

Glavne oblasti promena se mogu svesti na sledeće:

- Poboljšane performanse i novi 32-bitni moduli u sistemu. 3.11 sadrži 32-bitne veznike za mrežne kartice koji su u saglasnosti sa NDIS standardom (*Network Device Interface Standard* 3.0) i novim 32-bitnim sistemom za pristup datotekama. Ovo praktično znači da od mrežne karte, pa kroz mrežni protokol i mrežni klijent softver, do tvrdog diska u stanici postoji 32-bitna staza za podatke. Ovo drastično poboljšava performanse I/O preko mreže.
- Tesna integracija sa okruženjem nasleđenim od *Windowsa 3.1*. Komponente za umrežavanje su integralni deo *Windows 3.1* okruženja, svaka je implementirana kao veznik za virtualni uređaj (VxD) i bolje se slaže sa ostalkom *Windowsa* nego MS-DOS rešenja.
- Veće mogućnosti za administriranje i konfigurisanje u mreži. Deo paketa je alat za konfigurisanje ADMINCFG.EXE. Njime se može kontrolisati deljenje resursa na stanicama, kao i parametri bezbednosti za *Windows* stanice i one koje koriste *Workgroup Add-On for MS DOS*.



- Minimalno korišćenje konvencionalne memorije. Novi paket zauzima samo 4kB konvencionalne memorije kada se koriste 32-bitni veznici za mrežu. Ovo ostavlja više memorije za MS DOS aplikacije pokrenute u svojim virtuelnim 8086 mašinama.
- Puna podrška radа sa *Windows NT* i *Windows NT Advanced Server* sistemima. Verzija 3.11 je trenutno najbolji klijent za *Windows NT* i *Advanced Server*. Paket se ponaša kao klijent koji pruža potpunu 32-bitnu podršku za mrežne veznike ugrađenu u NT, kao i povećanu bezbednost u NT domenima. Pored ovoga, uz paket se isporučuje *Remote Access Service* (RAS) klijent za pristup modemom sa udaljene lokacije W4WG računarima ili *Windows NT* ili *Advanced Server* mreži sa instaliranim RAS serverom.
- Poboljšana podrška za *Novell NetWare* 3.11 može da radi sa ODI veznicima (*Open Datalink Interface*), podržava *peer* usluge preko IPX protokola i sadrži 32-bitni IPX/SPX kompatibilni transport koji se može rutirati. Ovim se omogućava da W4WG 3.11 radi na mreži sa IPX protokolom, uključujući i mreže na kojima se koriste IPX ruteri.
- Ugrađena bolja podrška za individualne računare. Tokom instalacije može se izabrati opcija da se ne instaliraju komponente za umrežavanje. I ovakva instalacija bolja je od standardnog *Windowsa 3.1* jer omogućava 32-bitni pristup datotekama i 32-bitni *cache*.
- Puna podrška za stari *Windows 3.1* stil umrežavanja. Verziju 3.11 možete instalirati preko starih veznika *Windowsa 3.1*. Time se zadržavaju dodatne funkcije nove verzije, ali bez korišćenja 32-bitnih mrežnih veznika.
- Ugrađene nove *Microsoft At Work* (MAW) tehnologije. 3.11 sadrži prvu implementaciju MAW tehnologije u vidu mogućnosti za slanje i prijem faks poruka.
- Konfigurisanje je pojednostavljeno jer se veznici više ne učitavaju iz CONFIG.SYS datoteke. NDIS veznici za mrežne karte i veznici za podršku W4WG 3.11 se učitavaju iz SYSTEM.INI datoteke. Iz CONFIG.SYS datoteke se pokreće samo jedan veznik (IFSMGR.SYS, podrška za 32-bitni pristup datotekama) i samo jedan iz AUTOEXEC.BAT.
- Poboljšane mrežne komponente. 32-bitni moduli (VREDIR, VSERVER) koji su postojali još u W4WG 3.10 su sada poboljšani. Podržano je slanje poruka na mreži, a *Print Manager* koji upravlja deljenim štampačem sada može stanice obaveštavati o statusu njihovih zahteva za štampanje. Poboljšanje komponenti za umrežavanje donosi poboljšanja delova sistema zaduženih za rad sa mrežom koji su kao 32-bitni uvedeni još u verziji 3.10. Pored ovoga, uvedeni su veznici za mrežne kartice zasnovani na NDIS 3.0 specifikaciji, koji obezbeđuju da ceo put podataka sa kartice do mrežnog usmerivača i servera bude potpuno 32-bitan. Ovim se primetno povećava brzina prenosa podataka preko mreže i smanjuje ograničenje brzine od strane samog *Windows* sistema.

U verziji 3.11 nisu zaboravljeni ni veznici za mrežne kartice prema 16-bitnoj NDIS 2.0 specifikaciji. U instalaciji nove verzije može se zadržati konfiguracija stare, urađene sa veznicima iz *Windowsa 3.10*.

Veznici za NDIS 2.0 su napisani u asembleru, to je 16-bitni kod za realni način rada procesora koji se izvršava u konvencionalnoj memoriji. Datoteke veznika obično imaju ekstenziju DOS. Svaki NDIS 2.0 veznik učitan u memoriju podržava samo jednu mrežnu karticu. Na primer, ako imate dve NE 2000 kompatibilne kartice u računaru, moraćete pokrenuti dve kopije veznika za nju, i dve kopije će postojati u memoriji. Protokoli pisani prema NDIS 2.0 specifikaciji zahtevaju korišćenje veznika prema istoj specifikaciji. S druge strane, NDIS 3.0 protokol mogu se koristiti i sa NDIS 2.0 veznicima, čime je ostvarena vertikalna kompatibilnost.

Ako je veznik napisan za NDIS 2.0, između njega i NDIS 3.0 protokola (na primer NetBEUI 386 ili NWLINK 386) potreban je sloj za mapiranje (NDIS2SUP.386) i stub za realni mod (NDISHEP.SYS) da bi se povezao protokol i veznik (proces koji se u literaturi naziva *binding*). Pored ovoga, NDIS 3.0 protokol zahteva i dodatni nivo za podršku (NDIS.386). Ovakva konfiguracija donosi izvesnu degradaciju performansi uz povećanu potrošnju memorije, jer se koristi prvi megabajt fizičke memorije da bi se moglo komunicirati sa veznikom u realnom modu.

NDIS2SUP.386 je sloj za mapiranje između NDIS 2.0 i NDIS 3.0 koji preslikava adrese generisane u realnom modu procesora u adresiranje potrebno za zaštićeni mod i obratno. Time se efektivno postiže prevođenje NDIS 2.0 naredbi u naredbe NDIS 3.0. NDIS2SUP.386 kreira međumemoriju za realni način rada u memoriji ispod 1MB koju će koristiti veznici za karticu.

Pre nego što se startuje *Windows for Workgroups 3.11* može se zaustaviti rad sa mrežom u realnom modu. To se postiže komandom NET STOP koja iz memorije uklanja 16-bitni NetBEUI protokol i usmerivač. Po pokretanju W4WG 3.11 koristiće se 32-bitni usmerivač, čime se postižu bolje performanse u saradnji sa ostalim 32-bitnim modulima za rad na mreži. Takođe, deljenje resursa nije moguće ako se koristi 16-bitni usmerivač.

32-bitni VEZNICI

Novost za *Windows for Workgroups* je da se u verziji 3.11, kao i kod *Windowsa NT*, podržavaju 32-bitni veznici za mrežne kartice zasnovani na NDIS 3.0 specifikaciji. Za pisanje NDIS 3.0 veznika koriste se 32-bitni C i C++ kod i funkcije API-ja. Specifikacija standarda obezbeđuje prenosivost veznika iz *Windowsa NT*: kod se može jednostavno modifikovati, a zatim prevesti za W4WG 3.11. NDIS 3.0 veznici izvršavaju se u zaštićenom režimu procesora i smeštaju se u produženu memoriju. Veznici napisani za NDIS 3.0 mogu se koristiti samo sa protokolima iz iste specifikacije.

U pravom NDIS 3.0 sistemu protokol stek se sastoji samo od tri sloja: NDIS 3.0-kompatibilnog veznika za mrežnu karticu, odgovarajućeg veznog sloja (NDIS 386) i NDIS 3.0 protokola.

Iz sistema se svi zahtevi za hardverske usluge iz NDIS 386 prenose na veznik za mrežnu karticu bez obzira na hardversku konfiguraciju sistema. Isto tako veznik na NDIS 386 prenosi zahteve hardvera za usluge sistema bez ikakvih znanja o operativnom sistemu. NDIS 3.86 je odgovoran za dodeljivanje memorije za zahteve koji zavise od hardvera, radnje pri podizanju i gašenju sistema i za obradu i čuvanje (*parsing*) parametara zadatih u datoteci PROTOCOL.INI.

NetBEUI 386 je virtuelni uređaj i NDIS 3.0 veznik za NetBEUI protokol. On omogućava da računar sa W4WG 3.11 komunicira sa mrežom. Protokol AsyBEUI za usluge pristupa mreži sa udaljene lokacije (RAS, Remote Access Services) je ugrađen u NetBEUI 386. AsyBEUI je sličan NetBEUI protokolu ali je namenjen područci sporih deonica mreže, kao što je asinhrona veza preko telefonske linije. *VNETBIOS je virtuelni uređaj koji obezbeđuje virtuelizaciju NetBEUI interfejsa za aplikacije u virtuelnoj mašini. To je interna funkcija, što je označeno zvezdicom ispred imena (*). Njegovo postojanje je neophodno za rad bilo kojeg virtuelnog uređaja koji koristi NetBIOS interfejs, kao što je VREDIR 386. Ovaj deo novog W4WG se u verziji 3.10 učitavao kao posebna datoteka VNETBIOS 386, dok je ovde ugrađen u WIN386.EXE.

VREDIR 386 je 32-bitni mrežni usmerivač (*redirector*). Obezbeđuje mapiranje lokalnih imena uređaja na udaljene mrežne uređaje. U njega su ugrađene i funkcije za pregled resursa mreže (*browse*). VSERVER 386 je server, 32-bitni virtuelni uređaj. Server obezbeđuje deljenje resursa lokalnog računara.

DODATNE MOGUĆNOSTI

Pored podrške za NDIS 2.0 i NDIS 3.0 veznike za mrežne kartice, W4WG 3.11 može da se izvršava i iznad *Open Datalink Interface* (ODI) veznika i IPX protokol steka. Uz NetBEUI protokol, ugrađena je podrška za novi NDIS 3.0-kompatibilan protokol – *NWLink*, koji je kompatibilan sa IPX/SPX protokolom.

Windows for Workgroups 3.11 sada podržava Microsoft TCP/IP protokol. To omogućava da sistem funkcioniše kao mrežni usmerivač i server na WAN mreži sa TCP/IP-om. Microsoft TCP/IP podržava i NetBIOS, pa je preko njega moguć i Network DDE. Usluge NetBIOS-a postoje i u IPX/SPX kompatibilnom transportu, kao 32-bitni virtuelni uređaj NWNBLink, koji pruža bolje performanse nego Novellov vlastiti rezidentni NETBIOS.EXE.

32-bitni pristup disku

Mogućnost 32-bitne komunikacije sa kontrolerom tvrdog diska, poznate i kao *FastDisk*, prvi put je predstavljena u *Windowsu 3.1*. Ovde je moram pomenuti jer je u vezi sa novim 32-bitnim pristupom datotekama u *Windows for Workgroups 3.11*.

FastDisk je skup 32-bitnih veznika u zaštićenom modu procesora 80386 ili višeg. Pozivi Int 13h koji se upućuju disk kontroleru se filtriraju i prosleđuju ili kroz 32-bitni interfejs, ili kroz sistemski BIOS. 32-bitni pristup disku radi direktno sa kontrolerom diska, a ne sa samim diskom. Podatke pristigle od kontrolera *Windows* može obrađivati po 32 bita (4 bajta). Između diska i kontrolera komunikacija je i dalje 16-bitna. Rutine koje se u *Windows*-u staraju o ovome zamenjuju delove BIOS-a zadužene za obraćanje disku.

Podržan je najveći deo kontrolera, ali je tu i „kvaka 22": za kontroler mora postojati odgovarajući virtuelni uređaj. *Windows 3.1* i W4WG 3.11 se isporučuju sa jednim takvim uređajem – WDCTRL, koji podržava sve kontrolere kompatibilne sa interfejsom preme Western Digital WD 1003 standardu. To je bio standard davno razvijen za prvobitni IBM PC AT i kojem i danas odgovara bar 90% instaliranih diskova. WDCTRL ne podržava ESDI i SCSI diskove, ali obično proizvođači za njih pišu posebne virtuelne uređaje. U vreme pojave W4WG 3.11 veznika za svoje kontrolere isporučivali su Future Domain, Ultrastor, Quantum i Compaq.

Korišćenje 32-bitnog pristupa disku je opcija koja je u instalaciji isključena. Microsoft vam je ostavio da je uključite sami iz **Virtual Memory** dijaloga u **Control Panel**-u. Sistem će raditi ako je i ne uključite, ali ćete biti uskraćeni za poboljšanje performansi koje donosi na većini sistema. *Setup* je napisan tako da je proveravanje kompatibilnosti sa WD-1003 vrlo strogo i naginje negativnoj odluci u slučaju makar i najmanjeg otklona od standarda. Ako se utvrdi da kontroler nije kompatibilan, polje *Use 32-bit Disk Access* će ostati zatamnjeno i nedostupno.

WDCTRL je projektovan za maksimalno čuvanje podataka. Testiranje kompatibilnosti ne obavlja se samo prilikom instalacije, već i pri svakom startovanju sistema. Prvo se probe sa čitanjem male količine podataka da bi se uvrdilo da li ponašanje kontrolera liči na ponašanje WD 1003, ako se nađe da liči, probaće se sa čitanjem veće količine podataka. Konačno, nakon par proba, WDCTRL će pokušati da „stvarno" pročita nešto sa diska. Ako to uspe, probaće da upiše neke podatke i pročita ih. Ako test čitaj/piši proba, WDCTRL će smatrati da može da se uključi i *Windows* će nastaviti sa radom.

Microsoft ne savetuje upotrebu *FastDiska* na prenosivim računarima sa ugrađenim sistemom za štednju električne energije (APM ili sličan). Kod njih se u trenucima neaktivnosti disk isključuje, a veznik za pristup disku to ne može da zna. Postoje, međutim, i moderni prenosnici koji su sasvim kompatibilni i sa ovim delom *Windowsa*, pa je na njima bezbedno koristiti i 32-bitni pristup disku.

Poboljšanje performansi korišćenjem 32-bitnog pristupa disku

Tri su područja na kojima se može primetiti brži rad sa diskom i poboljšanje ukupnih performansi: brže izvršavanje MS-DOS aplikacija, jer mogu dobiti više stranica virtuelne memorije za izvršavanje; brže straničenje za virtuelnu memoriju, jer se korišćenjem *FastDiska* može u memoriju učitati samo onoliko stranica koliko aplikacije zaista zahteva, pa je preklapanje između programa praktično trenutno, i, na kraju – ceo *Windows* radi brže, jer je i inače to veoma disk-intenzivan sistem.

Još jedna prednost 32-bitnog pristupa disku je mogućnost korišćenja preklapanja i ulaza i izlaza (*Overlapped I/O*), kada se zahtevi za čitanje ili pisanje na disk mogu obrađivati asinhrono. Zbog ovoga se straničenje može dešavati u pozadini dok se aplikacija normalno izvršava, a zahteve za operacije sa diskom može istovremeno izdavati više aktivnih programa.

Pristup disku iz MS-DOS-a

U MS-DOS-u put zahteva za korišćenje diska je sledeći: 1. Aplikacija šalje Int 21h poziv MS-DOS-u. 2. MS-DOS nalazi lokaciju podataka na disku prema informacijama u FAT tabeli (*File Allocation Table*). 3. MS-DOS poziva BIOS diska preko Int 13h. BIOS sadrži programe za kontrolu osnovnih uređaja u sistemu, pa tako i diska. 4. BIOS diska obraća se direktno kontroleru. Svaki kontroler ima svoj BIOS, ovi BIOS-i ne zavise od diska, već od ploče kontrolera ili adaptera koji povezuju disk sa ostatkom računara.

Ako je disk nekog posebnog tipa, proizvođač može da uz njega isporuči veznik koji presreće Int 13h i proverava da li će disk razumeti poziv. Obično ovakvi veznici preuzimaju sve zahteve za čitanje sa diska, pa poziv i ne dođe do ROM BIOS-a računara.

Iz Windowsa bez FastDiska

Stvari se u *Windowsu* koji radi u zaštićenom načinu rada dodatno komplikuju zbog potrebe za preklapanjem iz zaštićenog u realni način rada. *Windows* mora da se prebaci u realni način rada (ili virtuelni 8086 način) da bi mogao da izvršava kod MS-DOS-a i BIOS-a koji je pisan za realni mod procesora. Prebacivanje između modova troši vreme i slabi performanse (slika 1).

Kada MS-DOS aplikacija pod *Windowsom* u zaštićenom načinu rada pokuša da čita iz datoteke, to se dešava na sledeći način: prvo *Windows* pokrane aplikaciju u prozoru u realnom načinu rada. Aplikacija pozove Int 21h da bi pročitala datoteku. Zaštićeni način rada presreće ovaj interapt i prebacuje u deo koji radi u *protected* modu, gde nekoliko virtuelnih uređaja pregleda poziv i pokušava da vidi šta *ne-Windows* program želi. Ako poziv nije ni za jedan od 386 uređaja, vraća se MS-DOS-u. *Windows* se tada prebacuje nazad u realni način rada i pušta da MS-DOS obradi poziv. MS-DOS čita sa diska i na kraju aktivira Int 13h da bi se obratio BIOS-u. Dalje postupak ide kao kod standardnog pristupa iz MS-DOS-a, van *Windowsa*.

Iz Windowsa sa FastDiskom

32-bitni pristup disku smanjuje broj preklapanja između realnog i zaštićenog moda procesora, jer Int 13h obrađuje u zaštićenom načinu rada. *FastDisk* filtrira sve pozive za disk kontroler i šalje zahteve na kontroler zaobilazeći BIOS.

U procesu učestvuju sledeće komponente:

*Int13	Presreće i emulira Int 13h pozive koje BIOS-u upućuje aplikacija za kontroler diska. *Int13 prenosi ove pozive do BlockDev uređaja, koji ih filtrira i smešta u red za čekanje na pristup disku. Int13 i PageFile služe kao ulazni sistem za 32-bitni pristup disku.
*PageFile	Stara se o datotekama za straničenje virtuelne memorije. Kroz BlockDev šalje pozive kontroleru diska kada su mu potrebne usluge diska.
*BlockDev	Jezgro 32-bitnog pristupa disku. Deo koji kreira i upravlja redom za čekanje Int 13h poziva koji aplikacije upućuju disk kontroleru. Filtrira Read, Write i Cancel pozive i prosleđuje ih kontroleru, a ostale pozive šalje BIOS-u na dalju obradu.
*WDCTRL	Uređaj koji komunicira sa standardnim WD-1003 ili ST 506 kompatibilnim disk kontrolerima. Uređaj se instalira samo ako Setup zaključi da je kontroler kompatibilan.

Kada je uključen FastDisk, u datoteci SYSTEM.INI naći ćete sledeće linije u sekciji [386Enh]:

```
32bitDiskAccess=On
device=*blockdev
device=pagefile
device=*int13
device=*wdctrl
```

Ako koristite različit veznik za 32-bitni pristup, u liniji **device=*wdctrl** sa desne strane naći će se ulaz koji zavisi od instaliranog kontrolera. Najčešće se u zadnje vreme mogu sresti veznici za VESA Local Bus adaptere za IDE diskove.

32-bitni pristup datotekama

Ovo je pravi novitet *Windowsa for Workgroups 3.11*. Put kojim *Windows* dobija i manipuliše podacima dobijenim sa diska preko Int 21h je potpuno 32-bitan i sva obrada se vrši u zaštićenom načinu rada, umesto da Int 21h obrađuje preko MS-DOS-a, u realnom načinu rada.

Poziv Int 21h manipuliše MS-DOS FAT tabelom, od čijeg sadržaja zavisi način na koji će se podaci pisati i čitati sa diska za koji postoji FAT. Pored obrade Int 21h u *protected* modu, 32-bitni pristup disku obezbeđuje i 32-bitnu zamenu za stari *SmartDrive* disk *cache* program, koji je baziran na MS-DOS-u i realnom načinu rada. Ceo sistem 32-bitnog pristupa disku je implementiran preko dva veznika za virtuelne uređaje: VFAT 386 i VCACHE 386.

Kao i 32-bitni pristup disku, i 32-bitni pristup datotekama je isključen nakon instalacije. Možete ga uključiti iz **Virtual Memory/Change** dijaloga unutar **386 Enhanced** dijaloga u **Control Panel**-u.

Zahtevi kompatibilnosti

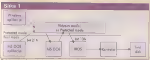
Pri učitavanju VFAT 386 prepoznaje fizički instalirane tvrde diskove u sistemu i podiže (*mounts*) svaki disk koji je podržan 32-bitnim pristupom disku. „Podizanje" diska se ovde odnosi na proces koji aktivira VFAT da bi saopštio sistemu da zahteve za I/O sa diskom usmerava na njega, a ne kroz lanac MS-DOS veznika.

Kao što MS-DOS za komuniciranje sa kontrolerom tvrdog diska koristi rutine iz BIOS-a, VFAT koristi 32-bitne *protected mode* komponente *FastDiska* da bi čitao i pisao sa diska. Da bi VFAT mogao da podigne disk, kontroler diska mora biti WD 1003-kompatibilan ili u sistemu mora biti instaliran modul za mapiranje u realni mod (*32-bit real mode mapper*).

Mapiranje u realni način rada

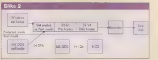
Postoji mnogo kontrolera koji nisu WD-1003-kompatibilni. Tipični primeri su SCSI i ESDI kontroleri. Da

Slika 1



Pristup tvrdom disku bez *FastDiska*

Slika 2



Pristup tvrdom disku sa uključenim *32-bit Disk Access* i *32-bit File Access*

bi podigao disk priključen na takav kontroler VFAT mora videti 32-bitni interfejs za pristup disku

Ako ne postoji veznik koji bi omogućio da se sa kontrolerom komunicira 32-bitno interapti za pristup disku (Int 13h) moraju se obrađivati u realnom načinu rada Posebni veznik za virtuelne uređaje (RMM D32) obezbeđuje mapiranja kojim se pozivi u zaštićenom načinu rada uzimaju od VFAT-a prevode u realni mod i šalju dalje kroz lanac MS DOS veznika. Mbeul za mapiranje se automatski instalira ako je uključen *32-bit File Access* a nije uključen *32-bit Disk Access*

Datoteka modula za mapiranje (RMM D32) se ne pojavljuje u [386Enh] sekciji SYSTEM INI datoteka već je učitava virtuelni uređaj VXDLDR 386 uređaj IOS 386 daje podršku za modul za mapiranje Kada je modul instaliran u SYSTEM INI datoteci će se naći sledeće linije

device=ifsmgr.386
device=ios 386
device=vxdldr 386
device=vfat.386
device=vcache.386
IFS Manager

Srce sistema za pristup datotekama u W4WG 3 11 je veznik IFS Manager On vodi računa o tabeli u kojoj se nalaze podaci o tipu uređaja za svaki instalirani disk i particiju i na osnovu toga odgovarajućem uređaju propušta zahteve za čitanje ili pisanje Kada aplikacija upotrebi poziv Int 21h IFSMGR ga proverava i poredi sa ulazima u tabelu instaliranih disk uređaja. Ako otkrije da je Int 21h upućen mrežnom disku proslediće interapt mrežnom usmerivaču (VREDIR 386) Ako je interapt za neki od diskova koje je podigao VFAT IFSMGR će poziv Int 21h propustiti VFAT u Na kraju ako nijedan veznik nije preuzeo poziv, Int 21h se šalje MS-DOS u

VFAT sve pozive Int 21h koji se odnose na njegove diskove obrađuje u zaštićenom načinu rada procesora. Zbog toga su performanse rada sa diskom bolje nego u običnom *Windowsu 3 10*

Kada je aktivan VFAT za obradu Int 21h koji je inicirala neka *ne-Windows* aplikacija potrebno je samo jedno prebacivanje iz zaštićenog u realni način rada ako se disku obraća *Windows* aplikacija nije potrebno nijedno prebacivanje (slika 2)

Saradnik VFAT a je veznik VCACHE 386 koji je odgovoran za rukovanje disk kešom za lokalne i mrežne diskova (VFAT 386 i VREDIR 386 respektivno) Sva keš operacije su u nadležnosti VCACHE-a uključujući dodelu memorije za keš oslobađanje memorije i algoritam za beređivanje kada „ostarele" podatke u kešu treba zameniti svežim podacima sa diska.

DOVIĐENJA SMARTDRIVE

Pojava 32 bitnog pristupa sistemu datoteka i 32-bitni disk keš su novosti *Windowsa for Workgroups 3.11* koje ćete odmah primetiti ako vam je poduda za prelazak na novu verziju bile poboljšanje performansi Pored obezbeđivanja obrada usluga Int 21h u zaštićenom modu VFAT u zajednici sa VCACHE-om obezbeđuje 32-bitni disk keš za *Windows* VFAT je zadužen za čitanje i pisanje na disk dok je VCACHE zadužen za rukovanje podacima koje VFAT piše ili su prisutne u keš memoriji Razlika u odnosu na *SmartDrive* su sledeće

- Rutine za 32 bitni pristup datotekama su napisane za zaštićeni način rada procesora pa se vreme ne gubi na preklapanjima iz zaštićenog u realni mod
- Rutine koje u 32-bitnom pristupu datoteka rade *read-ahead* (čitanje unapred) implementirane su na bazi čitanja celih datoteka a ne sektora pa je verovatnoća pogodaka u tom segmentu veće
- Rutine koje u sistemu za 32 bitni pristup datotekama rade *caching* dele keš memoriju sa mrežnim usmerivačem (VREDIR 386) pa je izbegnuto trošenje memorije za odvojene zone keš memorija
- Rutine 32-bitnog pristupa datotekama rade keširanje na bazi keširanja datoteka (*per file*) a ne sektora (*contiguous sector*) kao što radi *SmartDrive*

Mali podsetnik cache termina

Read-ahead caching – keširanje sa čitanjem podataka unapred pre nego što su traženi Koriste ga neki cache programi koji pokušavaju da predvide šta će sledeće biti traženo sa diska pa se podaci sa diska učitavaju pre nego što se od aplikacije stvarno primi zahtev Ako su podaci koji kasnije aplikacija zatraži već u kešu performanse su svakako unapređene Ako se međutim podaci ne nalaze u *read-ahead* kešu moraće se učitati sa diska Dva algoritma za čitanje unapred koji se često koriste su keširanja uzastopnih sektora (*contiguous sector caching*) i keširanja celih datoteka (*per file caching*)

Contiguous-sector caching ili keširanja uzastopnih sektora znači da se za svaki zahtev sa diska čitaju i dodatni uzastopni sektori prema pretpostavci da će sledeći zahtev tražiti podatke baš iz tih sledećih sektora. Ovakav tip keširanja dobar je za aplikacije koje barataju bazama podataka ili za neka druga situacija u kojima se podaci sekvencijalno čitaju Problem je u tome što algoritam pretpostavlja da je datoteka zapisana u uzastopnim sektorima Ovo u opštem (i vrlo čestom) slučaju nije tačno jer je fragmentacija diska sasvim normalna pojava Zato se sa učitavanjem uzastopnih sektora često učitavaju delovi nekih drugih datoteka pa se keš puni podacima koji ni su potrebni Ovim se znatno smanjuje verovatnoća pogodaka odnosno verovatnoća da će podaci koji se zahtevaju zaista biti u keš memoriji

Per-file caching znači da se posle zahteva za čitanje neke datoteke učitava samo deo datoteke (u sektorima) i da se pored toga unapred učitavaju dodatni sektori koji pripadaju istoj datoteci Pretpostavka je da će sledećim zahtevom aplikacija tražiti još jedan deo upravo otvorene i učitane datoteka Ovaj tip preranog čitanja je bolji od prethodnog jer se učitavaju samo delovi datoteka koja je zaista prvi put i zatražena pa se ne trpe posledice nastale zbog fragmentacije datoteka. Naravno fragmentacija na disku i dalje utiče na performanse ali u smislu dužeg vremena koje je mehaničkom delu diska potrebno da uzastopne sektora iste datoteka pokupi sa različitih lokacija na disku

Tehnika po imenu **write-behind caching** ili **lazy writing** (odloženo pisanje na disk) se takođe koristi u nekim programima za keširanje Podaci koji stižu za pisanje se čuvaju a stvarno pisanje se odlaže do trenutka u kojem je ceo računar manje opterećen Dužina intervala u kojem se pisanje odlaže je promenljiva ali nikada nije veća od par sekundi Kada keš izvesti da je sposoban za zakasnelo pisanje kontrola se vraća aplikaciji koja sada može mnogo brže da piše prema disku jer ne piše fizički na disk već u RAM memoriju Iako odloženo pisanje može efektivno poboljšati performanse rada sa diskom naravno da postoji i loša strana informacije se ne pišu na disk sve dok se keš ne „isprazni" (*flush*) Zbog ovoga je važno da se računar ne ugasi odmah nakon operacije u kojoj je neka aplikacija pisala u datoteku a pre nego što su podaci zaista upisani na disk

Suživot sa softverom za kompresiju

Primer potrebe za korišćenjem modula za mapiranje zaštićenog moda poziva VFAT-a u realni mod je podrška pristupu komprimovanom disku preko VFAT-a, i to disku komprimovanom pomoću softvera kakav je *DoubleSpace* iz MS DOS a 6 2 ili *Stacker* Pošto se svi podaci pre pisanja na disk kontroler moraju dodatno obrađivati komponenta za 32 bitni pristup disku ne mogu se koristiti U tom slučaju VFAT mora da se služi lancem MS-DOS veznika koji će obraditi Int 21h zahtev

VFAT neće moći da podigne disk koji je komprimovan *DoubleSpace*-om iz MS-DOS a 6 0 već zahteva da je softver iz verzije 6 2

Drugi problem sa komprimovanim diskovima je u vezi sa veličinom slobodnog prostora na takvom disku Keš mora pouzdano da zna koliko je mesta ostalo na disku znati to na komprimovanom disku je teško, jer se ne zna koliki će prostor zauzimati komprimovane informacije zbog činjenice da se stepen kompresi je razlikuje u zavisnosti od tipa podataka

Nakon instalacije je zakasnelo pisanje isključeno za sve komprimovane diskove koji su u nadležnosti VFAT a osim diskova komprimovanih pod *DoubleSpace*-om iz MS DOS-a U skupu *DoubleSpace* funkcija iz verzije 6 2 je i mehanizam koji VFAT može upotrebiti za ispitivanje veličine slobodnog prostora pa se može omogućiti i odloženo pisanje

Odloženo pisanje za druge vrste kompresionog softvera možete uključiti unošenjem parametra ForceLazyOn=1 u [vcache] sekciju SYSTEM INI datoteke

Kada je u W4WG 3 11 uključen *32 bit File Access*, kombinacija VFAT+VCACHE obezbeđuje funkcije koje je ranije obezbeđivao *SmartDrive disk cache* program Kada se pokrene takva kopija *Windows*-a *SmartDrive* se automatski isključuje za sve diskove koji se podižu preko VFAT a.

Memorija koju ste dodelili *SmartDrive*-u u MS DOS-u ostaće alocirana pa *Windows* neće moći da je koristi Zbog toga uz uključen *32-bit File Access* treba veličinu *SmartDrive* memorije pod *Windows*-om smanjiti na recimo samo 128KB

Nemojte, međutim potpuno napustiti *SmartDrive* i nemojte ga brisati iz AUTOEXEC BAT datoteke On još uvek zadržava svoju funkcionalnost za keširanje flopi i CD ROM uređaja CD ROM uređaji se keširaju uz pomoć veznika MSCDEX EXE a novi *SmartDrive 5 0* podržava i pisanje na CD-ROM i prepoznaje kada je u uređaju zamenjen disk

SCENARIJI ZA RAD SA DISKOM

Pristup disku koji ne podiže VFAT bez FastDiska

Ovo je *default* situacija koju srećete odmah nakon početne instalacije W4WG 3 11 Pozivi Int 21h se obrađuju u MS-DOS-u preko njegovih 16-bitnih veznika za *real mode* procesora U sekciji [386Enh] nalaze se sledeće linije

device=ifsmgr 386
device=*blockdev

Nastavak na strani 64

BORLANDOV KVANTNI SKOK

Nova verzija najpopularnijeg C++ prevodioca stigla je kod nas u svojoj punoj snazi, i zatekla nas potpuno nespremne – izmene koje su obuhvatile skoro sve delove paketa su do te mere temeljne da se na prvi pogled čini da se radi o potpuno novom proizvodu

Bojan Petrović

Borlandovi prevodioci su svoju slavu stekli pre svega na već čuvenoj integrisanoj okolini koja je u programiranje unela pravu revoluciju. Mnogo jače firme, u prvom redu Microsoft, su tek mnogo kasnije usvojile ovaj koncept, i nikada nisu uspele da postignu ono što je uspeo Borland – da ubede korisnike da je njihov pristup bolji.

Nova verzija *Borland C++* prevodioca stiže na više disketa, zauzima više megabajta prostora, zahteva više učenja i postiže bolje rezultate od svih svojih prethodnika, ali i većine takmaca. Napušteno je programiranje pod DOS-om osim sa komandne linije i u potpunosti se prešlo na IDE okolinu pod *Windows* sistemom. Uvedena je neka vrste *Visual* sistema, *Object Windows* biblioteka je drastično prerađena, *Expert* sistemi omogućavaju izuzetno lako pravljenje kostura aplikacija sa svim poznatim elementima modernih *Windows* aplikacija. Kompajler proizvodi kod za 286, 386, 486 i Pentium računare, za DOS 16-bitni i 32 bitni *Windows*, ali i za *Windows 4* (alias *Chicago*). Podržane su najnovije izmene u ANSI C++ specifikaciji, kao i RTTI i *Exception handling*. Editor, projekt menadžer i help su doživeli dalje sjajna poboljšanja, dodatni alati su bolji nego ikada. Jednom reču, *Borland C++ 4.0* predstavlja najbolji paket ove vrste trenutno na tržištu. Naravno, niko ne zna do kada će tako biti, jer svi čekaju sledeću verziju glavnog takmaca, *Microsoft Visual C*.

Novine i promene koje čine ovu verziju kompajlera su toliko opsežne da je sasvim nemoguće opisati ih sve, pa čak ni pronaći ih sve u prvih nekoliko dana rada. Mi ćemo ipak pokušati da pomenemo ona najinteresantnije, i unapred se izvinjavamo ako nešto propustimo.

INSTALACIJA

Kao i kod svih modernih aplikacija, instalacija je skoro bezbolna. Počinje standardnim **Run** iz **Windows File** menija, i dalje se od nas očekuje sasvim minimalan napor i nešto malo više vremena. Mada instalaciona procedura nudi podešavanja svega i svačega, počevši od direktorijuma, delova aplikacije koji će se instalirati, mogućnosti rada na LAN Windows sistemu, instaliranja WIN32s proširenja, pa do nove grupe za ceo paket, čini nam se da je sve tako fino uređeno da je dovoljno odabrati „automatsku instalaciju" da biste bili zadovoljni. Naravno, usput se precizno proverava da li za ono što se instalira ima mesta na disku, tako da možete biti sasvim sigurni da će sve biti kako treba. Proces traje, naravno, nezanemarljivu količinu vremena, pri čemu vas ugodni zvučni signali prekidaju u gledanju nekog filma da biste promenili jednu od dvadeset dve diskete, na koliko dolazi ovaj paket. Momci iz Borlanda su se zaista potrudili da ovaj postupak učine ne samo jednostavnim nego i simpatičnim. Sve vreme instalacije ekran simulira vožnju kolima kroz neki predeo američkog zapada, pri čemu „brzinometar" na instrument tabli pokazuje koliko smo zapravo kilobajta instalirali, a reklamne table koje promiču pored puta daju prve informacije o tome šta nas tek čeka. Dobro, ovo je već viđeno, ali ipak predstavlja pravi uzlet u odnosu na neka druga savremena „promeni disketu i čekaj" instalacije.

Trenutak pošto se završila instalacija, instaliran je i WIN32s modul koji uvećava snagu *Windows 3.1* sistema proširenjima za 32 bitne aplikacije, a zatim nova *Windows* grupa. Ovo je tek bio zanimljiv prizor za oči: broj ikona koje su se u grupi pojavile je zaista impresivan, čak zastrašujući. Samo pogled na njihov broj sugeriše da je ovde reč o veoma ozbiljnoj aplikaciji koja zahteva mnogo učenja. Sa druge strane, ovo je već pokazatelj da se očekuje veoma jak hardver, ma šta pisalo u dokumentaciji. A dokumentacija kaža:

- DOS 4.01 ili noviji
- *Windows 3.1* ili noviji u enhanced modu
- Hard disk od najmanje 40MB za minimalnu instalaciju ili 80MB za punu
- 1.44 flopi disk ili CD ROM (ukoliko ste došli do njega)
- Najmanje 4MB RAM
- *Windows* kompatibilan miš

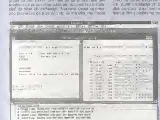
Ovo, međutim, predstavlja minimum minimuma. Par redova dalje pisac kaže da bi ipak trebalo imati 8MB RAM za komforan rad, koprocesor (ukoliko pišete programe koji koriste *floating point* matematiku), a ni bolji procesor od 386 ne bi smetao. Mi smo ovaj paket instalirali na 486DX/40 mašinu sa Quantum LPS270 diskom i lokal bas sistemom, pa nam se ipak činilo da bismo mogli da nađemo i jaču mašinu. Dakle, puna instalacija je zaista zauzela nekih 80MB disk prostora, dala nam vremena da pogledamo po neki film i pročitamo poneku reč iz dokumentacije.

PROLOG

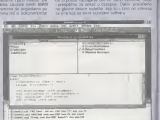
Prva važna činjenica jeste to da je u ovoj verziji napravljen veliki korak ka *Windows* programiranju, uz primetno napuštanje DOS okoline. Istina, još su tu verzije kompajlera za komandnu liniju, ali nema DOS IDE okoline, Turbo asemblera, Turbo profajlera. Deo dokumentacije namenjen programiranju pod DOS sistemom je sada izdvojen u knjižicu od nekih 150 strana, što je još jedan nagoveštaj da je vreme DOS aplikacija na izmaku. Upravo zato mi ćemo se skoncentrisati na *Windows IDE* okolinu, kao logičan izbor u ova prozorska vremena, i izvinjavamo se ljubiteljima DOS programiranja i komandne linije koje ćemo ovom prilikom malo zanemariti.

Možda je malo čudno, ali prvo što smo primetili ulazeći u IDE okolinu bio je nedostatak one lepe uvodne slike iz verzije 3.1. Nju je zamenila jedna mala, skoro zanemarljiva sličica, lišena likovnog smisla, ali sa vrednim objašnjenjem da upravo ulazimo u *Borland C++ 4.0*. Tada smo se našli u samom centru sveta IDE okoline koja je na prvi pogled izgledala gotovo kao i u prethodnoj verziji, uz manja dizajnerska poboljšanja. A onda smo malo bolje pogledali.

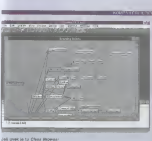
Integrisana okolina se i dalje sastoji iz standardnih celina, ali uz velike, čak dramatične izmene i poboljšanja. Editorski prozor i dalje izgleda kao stari, sve dok se ne upotrebi mogućnost deljenja prozora na korisne površine, i dok se ne obrati pažnja na njegove nova osobine i emulacije. Projekat više uopšte ne liči na stari, jer sada omogućava vizuelni prikaz međusobne zavisnosti datoteka u projektu pomoću neke vrste stabla čiji čvorovi se mogu širiti i skupljati. Pritisak na desni taster miša u bilo kom prozoru aplikacije otvara takozvani **Speed** meni, koji sadrži stavke interesantne za taj prozor i omogućava još jedan način da se do nekih opcija dođe lako i brzo. Debager više nije onaj nepijatni tekstualni iz prethodne verzije, već pravi integrisani debager sa svim standardnim mogućnostima. Više ne postoji **Compile** podmeni, već su se njegove opcije podelile na **Project** i **Debug** podmenije koji su se i inače promenili. Tu je i **Tool** meni, koji nudi lako pozivanje drugih aplikacija *Borland 4.0* sisteme, a pri tom mu se mogu dodavati nove stavke prema vašim željama. Ukratko, Borland IDE je pretrpeo ogromne promene u skoro svim aspektima, tako da bi prosto nabrajanje bilo ne samo zamorno, nego i preopširno za prikaz u časopisu. Dakle, prećićemo na glavne delove sistema, koji su i tako od interesa za one koji se bave razvojem softvera.



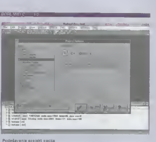
Novi izgled projekt menadžera i editorski prozor podeljen na površine



ClassExpert je zaista expert sistem za rad sa klasama



Još uvek je tu *Class Browser*



Podešavanje projekt opcija

EDITOR

Editorski prozori su i dalje na svu sreću oni u kojima stvaramo i razvijamo kod. IDE podržava do 32 otvorena editorska prozora u isto vreme, a među njima, naravno, i više puta otvorenu istu datoteku i slične kombinacije. I dalje su tu poznati **Status bar**, **Speed bar** (poznatiji kao **Tool bar**), **Undo** mogućnosti i tako dalje.

Prvo na spisku je ono što mnoge najviše i interesuje u vezi nove integrisane okoline: rad sa tastaturom i eventualna podrška *Brief* sistema. Pre svega, rad sa tastaturom je sada podržan kroz mapiranje tastature definisano preko *.KB datoteka sa opisima operacija za svaku akciju na tastaturi, korišćenjem makroa. Već postoje mape za podršku *Brief* i *Epsilon* sistema, kao i starog „Classic" mapiranja iz DOS verzije 3.1. Prema rečima tvorca, ove emulacije podržavaju većinu operacija inače dostupnih u njima, što će svakako obradovati korisnike naklonjene *Briefu*. Međutim, kako je ovde zaista reč o emulaciji, ljubitelji *Briefa* koji su se trudili da napišu *Brief* makroe za sve svoje omiljene operacije moraće ili da ostanu na njemu, ili da napišu verziju za makro jezik koga podržava makrokompajler za *.KB datoteke *Borlanda 4.0* (*Key Mapper*). Zanimljiva je i činjenica da su se tvorci setili da u **File/Open** meniju, osim standardnih CPP, C, H i ostalih „programerskih" ekstenzija, ponude otvaranje datoteka i sa KB ekstenzijama, što sugeriše mogućnost rada sa njima iz integrisane okoline. I zaista, ukoliko učitate jednu KB datoteku u editorski prozor i pozovete *KeyMap* kompajler u **Tool** meniju, odmah ćete dobiti potrebnu prevedenu verziju, bez izlaska iz IDE okoline.

Editorski prozori i dalje omogućavaju sintaksno bojenje sorsa, pri čemu se mogu zadati ekstenzije koje će diktirati koje datoteke treba farbati, a koje ne. Osim samog bojenja, podržani su i različiti fontovi, tako da komentari mogu biti *italic*, ključne reči *bold*, a ostali *normal*. Ovo zaista prijatno poboljšava čitljivost koda, ali malo i usporava proces ispisa na ekran. Međutim, lokalna magistrala uspešno rešava ovaj problem, pa eto prvog zahteva za poboljšanjem hardvera. Šema boja se, naravno, može menjati i to opet između nekoliko ponuđenih šema ili putem zgodnog dijaloga koji na svu sreću podseća na onaj iz prethodne verzije. Inače, od ponuđenih šema, nama se čini da je upotrebljiva samo „Default", pošto ostale imaju i malo neprijatne rasporede boja (zamislite svetlo plavu pozadinu sa plavim tekstom). Da ne zaboravimo, postoji i mogućnost zadavanja elemenata koji će biti podložni bojenju, opet putem datoteke sa spiskom tokena. Na taj način možete i neke svoje važne makroe ili funkcije proglasiti (što se bojenja tiče) službenim rečima i tako vizuelno pojasniti kod. Opet, da ta su dva moguća spiska, a na vama je da odlučite koji vam treba.

Rad u editorskom prozoru donosi nekoliko zanimljivih novina. Nama se najbitnijom učinila mogućnost deljenja prozora na korisne površine. Naime, moguće je podeliti površinu prozora na više manjih delova, od kojih će svaki predstavljati po jedan pogled na istu datoteku, tako da se ovim kompaktno eliminiše potreba za otvaranjem više prozora sa istom datotekom. Međutim, nekako nam se čini da je ove operacije zamišljena za one korisnike koji imaju sreću da rade sa 17-inčnim monitorima i/ili većom rezolucijom od 800x600, pošto se deljenjem prozora na površine njihove korisna površina zapravo smanjuje na neki deo originalne. Sa površinama se zatim može raditi zaista sve i svašta, počevši od očekivane promene veličine svake od površina, prebacivanje iz jedne u drugu, do daljeg deljenja ili gašenja.

Naravno, ima tu još dosta novosti, koje ćemo pokušati samo da pomenemo. Snimanje bekap verzija datoteke sada obuhvata ne samo sigurnosno snimanje prethodne verzije fajla, već i posebne **Mirror** i **Backup** direktorijume, gde prvi označava gde će se čuvati najnovija verzija promenjene datoteke, a gde rezervne kopije originala. Moguće je u trećem direktorijumu čuvati i originalni fajl, tako da su ovim tvorci obezbedili vaš rad od bilo kakvih problema. Dalje, editor omogućava emulaciju *Brief* kursora i regularnog pretraživanja, postavljenje desne margine (koja može biti i vidljiva), dubinu indentacije i tako dalje.

Jednom rečju, *Editor* je postao u pravi ozbiljan programerski editor, sa svim mogućnostima kojih su se tvorci dosetili. Naravno, reći će se *Brief* ljubitelji koji će reći da ovaj to odavno podržava, ali IDE je ipak IDE, zar ne?

PROJEKAT

Projekt menadžer je doživeo temeljne promene, čak možemo slobodno reći da je reč o potpuno novom pristupu temi. Prva stvar koja se primeti je da se projekt datoteke sada završavaju sa IDE, a da se stare projekt datoteke mogu koristiti, pri čemu će ih projekt menadžer „u letu" konvertovati. Projekat se sada posmatra na zanimljiv način – prema međusobnoj zavisnosti datoteke u projektu. To otprilike znači da možemo imati čvor projekta kao oca, čiji će sin biti EXE datoteka proistekla iz njega. Iz nje zatim slede sinovi koji su zapravo sors datoteke i sve ostalo što treba da se uključi u projekat. Svaki od tih čvorova dalje može da se reši na svoje pod datoteke, što su na primer H datoteke.

Dalje, projekt menadžer nudi izradu višestrukih projekata, što znači da pod jednom IDE datotekom kriju više srodnih projekata. Primer toga je dat u obliku projekta kojim se mogu istovremeno napraviti dve verzije WHELLO programa, jedna 16-bitna a druga 32-bitna. Naravno, moguće je raditi i sa jednim projektom višestrukog projekta, i sa svima odjednom.

Rad sa projektom obuhvata dodavanje, brisanje, premeštanje čvorova, kopiranje, promenu atributa. Atributi čvorova zapravo predstavljaju nadogradnju onoga što se tek nagoveštavalo u prethodnoj verziji IDE okoline. Atributi čvorova pre svega obuhvataju ciljne (*target*) atribute, koji se zaista lako menjaju upotrebom sjajnog **TargetExpert** dijaloga. On omogućava lak i brz izbor platforme za koju se prevodi: *Windows 3.0* ili *3.1*, *Windows 32*, DOS standardni, DOS overlay. Lako se menja i tip nastalog entiteta: standardna EXE aplikacija, DLL, statična LIB biblioteka, *Windows Help* datoteka i tako dalje. Da naglasimo, do **TargetExpert** a se lako dolazi pritiskom na desni taster miša u okviru projektnog prozora, čime se aktivira **SpeedMenu**, sjajna stvar za brzo i lako pozivanje čestih opcija. Dalje, atributi čvora obuhvataju i opcije prevođenja, što se menja aktiviranjem **Edit Node** stavke u **Speed** meniju. Ovo obuhvata ime čvora, opis, prevodilac i tako dalje. Izbor prevodioca je jako zanimljiv zbog koji nagoveštava mogućnost ručnog izbora načina obrade svake datoteke u projektu, a možda i dodavanje novog prevodioca spisku postojećih.

Na kraju, pomenućemo postojanje *Source Pool* sistema. To je, kako tvorac kaže, kolekcija čvorova koji se ne prevode, ali se referenciraju tokom prevođenja. Ovo omogućava, recimo, prevođenje istog sorsa u dve ciljne aplikacije, jednu 16-bitnu i drugu 32-bitnu. U *Pool* bi mogli da smestimo sve heder datoteke projekta, koje bi inače rasporedili na dva projekta, a dodavanje nove heder datoteke u projekat bi se svelo na njeno ubacivanje u pool.

Ugalvnom, projekt menadžer je postao visoko profesionalan, komplikovan sistem za rad sa velikim (ali i malim, naravno) projektima, koji omogućava gotovo sve što biste mogli da smislite. Čini nam se da su momci iz Borlanda malo i preterali, pošto se raznovrsnost mogućnosti i sjajne osobine ovog dela *Borlanda 4.0* delimično negiraju vremenom potrebnim za savladavanje svih tih veština.

KOMPAJLER

Aplikacije se mogu prevoditi korišćenjem IDE okoline, BCC kompajlerom za komandnu liniju za 16-bitne aplikacije, ili BWCC kompajlerom za komandnu liniju za 32-bitne aplikacije. Opet se izvinjavamo ortacima koji vole red sa komandne linije, ali IDE okolina je glavna osobina *Borland C++ 4.0* paketa i predstavlja pogled u budućnost.

Pre svega, osnovni postupci su ostali isti, samo se meni opcije nalaze u drugim podmenijima. **Project** meni sada sadrži **Compile**, **Make** i **Build**, dok je Run u Debug podmeniju. Tok prevođenja je isti kao i ranije, osim što je, čini nam se, malo sporiji (što je lako opravdati novim mogućnostima). Pri prevođenju jedne jako velike aplikacije nam se nekoliko puta desilo da kompajler prijavi nedostatak memorije (na mašini sa 8MB RAM), što se pokazalo kao lako prebrodivo sa boljim podešavanjem opcija prevođenja. A opcije prevođenja su smeštene (opet promenjeno) u **Options/Project** dijalog i nude zaista sve i svašta.

Tu su klasična opšta podešavanja kompajlera koje se skoro nimalo ne razlikuju od prethodne verzije. Jedino što se može primetiti je promenjen rad sa prekompajliranim heder datotekama. Ekstenzija više nije SYM nego CSM, ali je moguće i zadati neko svoje ime. Takođe, moguće je zadati heder fajl posle koga se dalji heder fajlovi neće ubacivati u CSM fajl, što

prema režime tvorca može da doprinese bržem i lak šem radu. Ovo na žalost nismo uspeli da potvrdimo.

Dalje slede opcije 16-bitnog prevođenja koje nu rad sa 286, 386 i 486 instrukcijama. 32 bitni kompajler obuhvata 386, 486 i Pentium, što nas je prijatno iznenadilo (mada smo zapravo i očekivali Pentium u spisku). Opcije specifične za C++ su proširene obradom izuzetaka (*Exception handling*) i RTTI (*Run time type information*). Ovo su istovremeno dve najvažnije nove osobine C++ prevodioca koje svakako treba pomenuti.

Najnoviji dodaci ANSI C++ standardu, RTTI i rad sa izuzecima, su u punoj snazi ukomponirani u *Borland C++ 4.0*. RTTI omogućava pisanje portabilnog koda koji može da odredi tačan tip objekta po startovanju, čak i kada se poseduje samo pointer ili referenca na objekat. RTTI omogućava i proveru tipa objekata i poređenje dva objekta po tipu korišćenjem TYPEID operatora. Tu je i nova klasa **Type_info**, koja uvodi „preopterećene" operatore == i != baš radi poređenja tipova objekata. Prost primer iz dokumentacije nagoveštava snagu ove novine:

```
char C;
float X;

if (typeid( C ) == typeid( X ) )
    cout << „C i X su istog tipa!" << endl
else
    cout << „C i X nisu istog tipa!" << endl
```

Obrada izuzetaka omogućava softversko reagovanje na pratnju abnormalnog završetka programa. Ovo sada može obuhvatiti prikupljanje informacija o uzroku „pada" sistema, smeštanje informacija kao i izvršavanje nekih operacija koje bi morale da se urede pre izlaska iz aplikacije. Uslov za sve ovo je da uzrok pada bude unutar aplikacije, što znači da se **Ctrl-C** ne uzima u obzir. Ova izvanredna mogućnost veoma unapređuje sigurnost sistema, olakšava otkrivanje grešaka i moguće eliminiše posledice nekih padova koji bi mogli biti opasni za sistem.

Optimizacija kompajlera su u osnovi ista kao u prethodnoj verziji. Prema našim merenjima nikakvi spektakularni dobici se ne mogu očekivati, mada, ruku na srce, nismo imali prilike da se oprobamo sa Pentium procesorima. Vreme prevođenja je teško oceniti, ali nam se čini da je sve ipak malo sporije nego ranije. Međutim, to se može objasniti velikim opterećenjem sistema, pošto nova IDE okolina nije iz gleda predviđena za svaku mašinu. Prvo smo se sreli sa porukom o nedostatku memorije pri završnoj operaciji linkovanju, što smo lako izbegli isključivanjem debug informacija za nekoliko datoteka projekta. Prekompajlirani hederi i dalje rastu do neslućenih veličina, ali i dalje znače strahovite dobitke za iole veće projekte. Ukratko, kompajler je kakav i treba da bude, stabilan i vertikalno kompatabilan.

Mada bismo mogli da kažemo da smo se iznenadili malim napretkom kompajlerskog dela sistema, moramo priznati da više i nismo očekivali. Stvar je u tome što je kompajler *Borlanda C++* već odavno prošao prave testove zahvaljujući dugom boravku na tržištu u vidu nekoliko prethodnih verzija i zaista ogromnom broju korisnika. Ova verzija je samo ispravila par poznatih bagova iz prethodne verzije i dodala podršku najnovijim elementima ANSI C++ standarda, što je i najznačajnije, naravno tu je i podrška Pentium procesora čiju svrsishodnost nismo bili u mogućnosti da proverimo.

DEBAGER

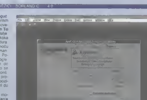
Najneprijatniji deo prethodne *Windows IDE* verzije bio svakako nedostatak integrisanog debagera. Ono što je postojalo je teško moglo da posluži svrsi i pri razvoju ole ozbiljnijeg projekta je izazivalo više neprijatnih misli u pravcu tog dela sistema. Ovaj put, međutim, tu je pravi integrisani debager koji konačno donosi raskoš nekadašnjeg DOS IDE debagera.

Sada imamo na raspolaganju standardne mogućnosti debagera *stepping* i *tracing*. Prva „koračanje" omogućava izvršavanje programa korak po korak, ne ulazeći u način izvršavanja tog koraka. To znači da će se uvek izvršiti po jedna linija sorsa, dok će se eventualni pozivi iz nje uraditi intamo bez prelaska u pozvane procedure. Drugi način kretanja je sofisticiraniji, pošto omogućava detaljno praćenje toka rada, uključujući i praćenje svih poziva i skokova koje bi „koračanje" preskočilo. Debager podržava i OBJ datoteke prevedene ranije sa debag informacijama u njima, čak i ako su proizvedene različitim jezicima, uključujući C, C++, Pascal i asembler.

Naravno, postoji mogućnost preskakanja velikih delova koda pri stepovanju korišćenjem **Run To Cursor** komande. Dalje, imate mogućnost skoka do željene procedure koju pronalazite pomoću **Locate Function** komande **Search** menija. Ponekad je korisno pogledati kako se došlo do trenutne pozicije, što se lako može izvesti korišćenjem **Call Stack** prozora koji pamti sve pozive koji su vas doveli do sadašnje pozicije.

Sledeće važna mogućnost je pregledanje promenljivih, izračunavanje izraza i izmene vrednosti podataka. Omogućeno je i formatiranje ispisa u **Watch** prozoru, čime se dodatno pojačava skup informacija koje se odatle dobijaju. Moguće je postavljenje prekidnih tačaka korišćenjem **BreakPoints** prozora i odgovarajućeg dijaloga kojim se precizno postavljaju osobine prekidne tačke. Pojedine tačke se mogu dodeli ili obrisati, isključiti na neki par od pogledati kod oko te tačke, pa čak i vršiti promene na njemu. Konačno, integrisani debager hvata i GP greške koje mogu nastati u vašem programu, pri čemu će vas pozicionirati na mesto nastanka. U tom trenutku bi trebalo izabrati *Program/Terminate* kako bi se sprečilo stvarni krah programa i ispravila nastala greška.

Jednom rečju, *Windows IDE* debager je sada baš onakav kakav je i trebalo da bude, bez uočljivih bagova ili problema. Prave stvari za *Windows* programera pa i one druge.



AppExpert sistem za brzo pravljenje OWL aplikacija

OWL I EXPERT SISTEM

Prethodna verzija *Borlanda C++* je među programere unela pravu pometnju svojom *Object Windows* bibliotekom. Reč je o starom sindromu: kada se pojavio *Windows*, veoma mnogo korisnika je odbijalo da ga koristi iskreno verujući da je komandna linija prava stvar (slično se desilo sa pojavom automatskih menjača u kolima, ali je to druga priča). Kada se kasnije većina *Windows* programera navikla na direktan rad sa *Windows API* funkcijama, a neko im preporučio zaobilazak svega toga korišćenjem gotove biblioteke, prezrivo su odmahnuli — to nije prava stvar. Međutim, izgleda da vreme čini svoje – od prethodne skromne biblioteke sa mnogim nedostacima i problemima, OWL je prerastao u izuzetno ozbiljan sistem za brz i efikasan razvoj modernih i profesionalnih aplikacija. Kažemo „sistem", zato što nije samo reč o biblioteci već i o novim „vizuelnim" alatima koji se mogu koristiti u radu.

Broj i opseg promena u OWL biblioteci se najbolje može videti iz dokumentacije – dok je ranije postojala samo jedna knjiga od nekih 500 strana, sada stižu dve od ukupno 1000. Nova biblioteka nije kompatabilna sa prethodnom ne samo zbog izbacivanja *dispatch* tabela koje su u prethodnoj verziji uvedene radi izbegavanja silnih **Switch** petlji kod obrade poruka, već su i same klase drastično drugačije. Ovaj put je sve urađeno u potpuno čistom ANSI C++, što po rečima tvorca obezbeđuje portabilnost između 16-bitnih i 32-bitnih aplikacija. Uveden je *message cracking*, razdvajanje *Windows* poruka na skup informacija koje se šalju funkcijama za obradu u vidu klasičnih ulaznih parametara, umesto starog, rogobatnog slanja referenci na strukture. Tu su klase za *Tool bar*, *Status bar*, rad sa štampačem, enkapsulacija *Windows GDI* objekata i tako dalje. Većina klasa sada sadrži obrađene skoro sve bitne *Windows API* funkcije, tako da više ne mora da se brine o raznim parametrima tipa *Handle*, pošto ih klasa sama obrađuje. Uveden je novi način rada sa podacima, **Doc/View** sistem, koji na prvi pogled predstavlja malo previše komplikovanja, ali donosi i neke zgodne novine. Naime, tvorci su se setili da treba razdvojiti podatke koji se koriste kao izvor od samog njihovog prikazivanja. Na osnovu toga stvorili su **Doc** klase, čija je uloga da isključivo rade sa podacima u smislu njihovog prikupljanja, skladištenja i čuvanja. Za prikaz ili obradu podataka služe **View** klase, koje omogućavaju neograničen broj načina obrade istih podataka smeštenih u jednu **Doc** klasu. Jednostavno rečeno, jedan **Doc** objekt čuva sadržaj datoteke, a nekoliko **View** objekata se brine o ispisu na ekran u vidu editorskog prozora, ispis u vidu statističkih podataka, ispis u vidu pregleda sadržaja itd.

Najinteresantnije za one koji su već radili sa OWL 1.0 bibliotekom je mogućnost prenošenja programa pravljenih sa njom na novu verziju. Na žalost, ovo će biti teško izvesti, ali moguće. Nekih pedeset strana knjige *Programmers Guide* je posvećeno postupku konvertovanja, koji zamislite može da se učini na dva načina. Prvi, naravno, je iz IDE, dok drugi, opet naravno, radi sa komandne linije. U oba slučaja morate prethodno izvršiti niz pripremnih operacija koje obuhvataju kopiranje prethodne verzije biblioteke u strukturu direktorijuma *Borlanda C++ 4.0*, manju preradu dela sorsa biblioteke, njeno prevođenje novim kompajlerom, uvođenje **Strict** promenljive u vaš sors radi boljeg otkrivanja grešaka, prevođenje sorsa aplikacije pomoću tako pripremljene stare biblioteke. Potom, iz više puta pomenutog **Speed** menija pozivate *OwlCvt* aplikaciju koja skoro automatski konvertuje sors. Međutim, sve i nije tako lako kao što kaže tvorac. Posle niza manjih problema oko prevođenja stare biblioteke ustanovili smo da je skoro nemoguće konvertovati ole veću aplikaciju zbog nedostatka memorije. Jednostavno, konvertor treba koristiti sa komandne linije. Dalje, i pored „automatskog" konvertora, čeka vas oko 10 strana objašnjenja onih stvari koje nisu mogle biti tako prerađene, već ćete vi morati da smislite bolji način za njihovo popravljenje, bilo uz pomoć uputstava iz knjige, bilo uz pomoć magije. Ovo nas je jako razočaralo, mada potpuno razumemo razloge za probleme. Nekako smo smatrali da će se momci iz Borlanda, u skladu sa kvalitetom nove biblioteke, potruditi da prelazak urade daleko bezbolnije. Jedino pozitivno ovde je postojanje zaista detaljnih objašnjenja za svaki problem koji može da se pojavi pri konvertovanju, bez čega bi postupak bio zaista nemoguć. Mada je i ovako vrlo naporan.

Umesto da se sada bavimo klasama koje donosi OWL 2.0, možda je bolje da pomenemo **Expert** sistem. Naime, klase su toliko brojne, toliko opsežne i toliko gladne za dokumentacijom da smo sigurni da bi bile potrebna bar serija članaka da bi se njihove moći opisale. Sa druge strane, **Expert** sistem omogućava baš kao što bi mogli da zamislimo izuzetan način za stvaranje kostura aplikacije za *Windows* sa svim modernim atributima. Prvi deo, **AppExpert**, omogućava stvaranje jednostavnijih aplikacija baziranih na OWL 2.0 korišćenjem popularnog vizuelnog pristupa. Proces počinje pozivom **AppExpert** stavke u **Project** meniju, čime se ulazi u osnovni dijalog. Po izboru naziva projekt fajla ulazi se u **AppExpert** dijalog koji nudi podešavanje svih osnovnih atributa aplikacije: tip aplikacije, postojanje **Tool** i **Status** bara, podršku štampanju i help sistemu, osobine glavnog prozora, osobine potprozora, nazive sors fajlova, klase koje će se pri svemu koristiti i tako dalje. Ceo postupak je podržan stalnim pomoćnim objašnjenjima i

NASTAVAK NA STRANI 102

MODEMI

PROFESIONALNI



MODEMI

MODELI	PREPORUKA CCITT	BRZINA b/s	NAČIN PRENOSA		OPSEG		Iznajmljeni		komut.	NADGLEDANJE/TESTIRANJE		
			ASIN	SYN	OSNOVNI	TLF KANAL	2ž	4ž	2ž	LAP	LDP	511
PP 1200 GS	V-23	0-1200					HDX	FDH	HDH	*	*	
PP 2400 GS	V26/V26 BIS	1200/600					HDX	FDH	HDX			
PP 19,2 KOS		19.2k/9.6k/ 7,2k/4,8k/2,4k		*	*		HDX	FDH		*		

FDH — duplex HDX — poluduplex LAP — lokalno analogna petlja LDP — lokalna digitalna petlja

U proizvodnom programu VF d d — Fabrike za proizvodnju visokofrekventnih uređaja nalaze se već nekoliko godina modemi i prateća oprema iz oblasti prenosa podataka. Oni su rezultat samostalnog razvoja stručnjaka Instituta „Mihajlo Pupin" i tehničko-tehnološke proizvodne transformacije i standardizacije koju su izvršili stručnjaci Fabrika VF uređaja. Na taj način proizvedeni su modemi koji su u potpunosti plod domaćeg razvoja i u velikoj meri domaće tehnologija

Sledeći najnovije svetske tendencije u:

* razvoju µp i signal procesora
* poboljšanju dijagnostičkih i test funkcije
* jednostavnosti održavanja
* računarsko projektovenje štampanih veza
* primena DS hibridnih kola
* automatizovan proces kontrole kvaliteta

uz visok industrijski standard proizvodnje

Institut „Mihajlo Pupin" i VF d d realizovali su novu generaciju modema kompaktne konstrukcije, velike gustine pakovanje i male potrošnje. Savremenim rašenjima, dizejnom, primenjenom tehnologijom i ostverenim perfomansama modemi nove generacije uvrštavaju se u sam vrh savremanih komunikacionih uređaja na svetskom tržištu.

VF d.d

Preduzeće za proizvodnju
visokofrekventnih uređaja, Beograd
11080 Zemun Cvetna 4
Telefoni
Plasman (011) 619-585
(011) 617-757 611-611,
Telefax 109-454

SAMO IME GOVORI

Majkrosoftov pakat *Excel for Windows 5.0*, u duhu imena koje nosi, proglašen je za najbolji spredšit program 1993. godine. Nova verzija nudi mnoštvo novina i unapređenja, od onih koja se podrazumevaju, kao što je, najzad, rad u trodimenzionalnoj tabeli, pa do sasvim osobitih, kao što je mogućnost dorade aplikacije pomoću programskog jezika *Visual Basic*. Uz zaslužene pohvale Borlandovom *QuattroPro* i u očekivanju odgovora iz Lotusovog tabora, konstatujemo da je u ovom trenutku teško zamisliti bolji *Windows* program za radne tabele od *Excela*.

Vladimir Stamenović

Excel 5.0 je veliki u svakom pogledu. Naime, paket sa disketama sadrži 4 knjige sa uputstvima za upotrebu, koverat sa licencnom karticom i uputstvom kome se treba obratiti za pomoć ako nešto ne ide kako treba, kao i 9 (daval!) disketa od 3.5" 1.44 Mb, a kompletno instaliran zauzima čitavih 25 Mb prostore na disku. No, ne morate vršiti kompletnu instalaciju već možete izdvojiti samo ono što vam treba a kasnije po potrebi do davati i ostale delove.

Od hardvera neophodno je da imate IBM ili kompatibilni 80286 PC računar ili bolji, 3.5" ili 5.25" flopi disk, čvrsti disk sa minimalno 8 Mb slobodnog prostora (za minimalnu instalaciju najmenje 4 Mb RAM-a), grafički displej kompatibilan sa MS Windows-om verzija 3.1 ili novijom. Od softvera trebaće vam DOS 3.1 ili noviji, MS Windows 3.1 ili noviji u standardnom ili proširenom modu.

Uputstva za upotrebu su uobičajena za ovakvu vrstu programa: *User's Guide* (korisnički vodič) sa detaljnim objašnjenjima o programu na ukupno 786 strana, *Query User's Guide* sa uputama za korišćenje i organizovanje podataka iz različitih izvora (*Excel, FoxPro, dBASE* itd.) na 168 strana, *Visual Basic User's Guide*, nešto novo i zanimljivo, naime, možete automatizovati, dotarati i programirati *Excel* i u *Excel*-u koristeći *Visual Basic Programming System Aplication Edition* (knjiga ima 344 strane), *Quick Reference*, gde je na 50-ak strana dat kratak pregled mogućnosti programa.

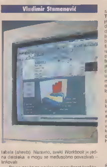
Dakle, ko reši da „razbije" ovu verziju *Excel*-a imaće posla sa oko 1350 strane teksta u priručnicima. Naravno, podrazumeva se da imate neka osnovna znanja o DOS-u i *Windows*-u.

Instalacija programa je uobičajena za ovu vrstu programa. Pokreće se instalacioni program iz *Windows*-a, koji „pretrese" diskove da vidi da li ima dovoljno mesta za program, a nakon toga na vama je samo da ubacujete diskete koje program traži. Preporučujem, zbog dužine trajanja instalacije, da se prvo uradi minimalna (laptop) instalacija, a kasnije možete dodati delove *Excel*-a koji vam trebaju. Minimalna instalacija traje oko pola sata. Pretpostavljam da bi to na nekoj novijoj mašini išlo dosta brže.

LEPOTA INTERFEJSA

Nakon završene instalacije treba resetovati računar i tek onda iz *Windows*-a pokrenuti *Excel*. Samo pokretanje programa je trajalo prvi put veoma dugo, čak oko 4 minuta! Prosto neverovatno. Međutim, malim eksperimentisanjem i dodavanjem disk keša stvar se znatno popravila, tako da je sada, nakon startovanja, program spreman za upotrebu posle 40-tak sekundi. Dakle, mogu da preporučim da, ako je to moguće, namestite minimum 1 Mb disk keša.

Dve krupne stvari karakterišu ovu verziju *Excel*-a. Prva je da najzad i u *Excel*-u imamo mogućnosti da radimo u trodimenzionalnoj tabeli. Ona se u dokumentaciji zove *Workbook*, u programu kratko – *book* (knjiga), a sastoji se od 256 tabela (*sheets*). Naravno, svaki *Workbook* je jedna datoteka, a mogu se međusobno povezivati – linkovati.



Druga značajna novina je mogućnost korišćenja *Visual Basic*-a za poboljšanje aplikacije, što je dosta bolja varijanta od Lotusovog *Add in Tool Box*-a.

Nakon startovanja programa, prvo što vidite je „knjiga" *book* koja se, kao što rekoh, sastoji od 256 tabela (*sheet1 – sheet256*). Interfejs je veoma lep i, kako se kasnije ispostavilo, izuzetno funkcionalan.

Book nudi šest tipova tabele:

- *Worksheet*, redna tabela za unos i manipulaciju podacima
- *Chart sheet*, dijagrami koji nisu povezani sa radnom tabelom
- *Visual Basic* modul, *Microsoft Excel Visual Basic* jezik
- *Dialog*, *Microsoft Excel Visual Basic* jezik
- *Microsoft Excel 4.0* radne tabele, kompatibilne sa tabelama te verzije
- *Microsoft Excel 4.0 international macro sheet* tabela koja obezbeđuje kompatibilnost sa prethodnim verzijama *Excel*-a

Neki detalji u interfejsu su isti kao u *Quattro Pro for Windows 5.0*, a tu pre svega mislim na način označavanja pojedinih tabela. Naime, na dnu tabele su izrazi kao u rokovniku, na kojima je oznaka broj tabele-stranice (*sheet1, sheet2,* ...). One se, naravno, mogu menjati, odnosno pojedinim stranicama dati nazivi po želji. Iznad tabele su postavljene četiri reda za razne namene.

U prvom redu je meni sa komandama za rad sa programom (**File, Edit, View, Insert, Format, Tools, Data, Window** i **Help**). Kao što je uobičajeno, meni je u padajućem stilu, a većina komandi ima svoje prečice (na primer Ctrl+C za komandu **Copy**). Mnoge od njih ili njihove potkomande (opcije) imaju lepo dizajnirane i jasne dijalog boksove preko kojih se mogu izvršiti razna podešavanja. Zanimljivo je i korisno i to što je princip rada u tabeli uglavnom ovakav: mišem označite celije ili oblast ćelija sa kojima nešto radite, pa tek onda pozivate željenu komandu ili više njih jednu za drugom, koje će delovati na označene celije. U nekim slučajevima ćete dobiti na ekranu u tabeli mali boks sa nekoliko ikona pomoću kojih možete brzo izvršiti različita podešavanja na označenoj oblasti ćelija. No, to ne mora biti pravilo, ali povoljno utiče na produktivnost.

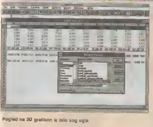
U drugom i trećem redu iznad tabele su smeštena dva tulbara, odnosno dve linije sa ikonama koje pomažu i ubrzavaju posao. S obzirom da već definisanih, gotovih ikona ima jako mnogo, a možete praviti i dodavati sopstvene, trebaće nešto vremena da naučite šta koja znači. U prvom momentu korisno je i originalno rešenje da kada postavite pointer miša na bilo koju od ikona, odmah dobijate ukratko njenu funkciju. Tu je i Help u kome možete naći detalje.

DUHOVITI SAVETI

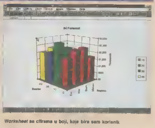
Prvi tulbar se zove standardni i u njemu su ikone pomoću kojih možete otvarati datoteke, snimati ih na disk, štampati, sortirati podatke u obe smera, kreirati grafikone na osnovu označenih podataka itd. Interesantna je ikona sa slikom sijalice koja, kada svetli, ima za vas neku informaciju ili pomoć koja se odnosi na posao koji trenutno radite. Taj sistem se naziva **TipWizard**. Ako kliknete na tu ikonu, iznad tabele, u dva reda ćete dobiti informaciju ili pomoć, a ako zatim kliknete na ikonu za **Help** (slika znaka pitanja sa strelicom), dobićete detaljne informacije. Vrlo zanimljivo i korisno, a ponekad su saveti **TipWizard-a** i šaljivi.

Drugi tulbar se naziva *Formatting toolbar* i sadrži 20-ak ikona za razna podešavanja u tabeli, kao što su izbor fontova i njihovih veličina, zadebljana, iskošena i podvučena slova/tekst, poravnavanje, zadavanje različitih formata za prikaz brojeva itd.

Sve u svemu, *Excel 5.0* nudi 13 različitih tulbarova koji se koriste zavisno od potrebe, odnosno posla koji obavljate: *Standard* – standardni poslovi u tabeli; *Formatting* – formatiranje karaktera, brojeva i ćelije; *Query And Pivot* – rad sa MS Query dodacima; *Chart* – kreiranje i modifikacije grafikona i slika; *Drawing* – kreiranje grafičkih objekata; *TipWizard* – korisne informacije za vreme rada; *Stop Recording* – zaustavljanje snimanja makroa; *Full Screen* - povratak u normalno stanje posle zuma; *Forms* – kreiranje *Visual Basic* dijalog boksova; *Visual Basic* – rad u *Visual Basic* modulima; *Auditing* – traženje veza, zavisnosti i grešaka u formulama; *Workgroup* – traže-



Pogled na 3D grafikon iz bilo kog ugla



***Worksheet* sa ciframa u boji, koje bira sam korisnik**

nje prijem i slanje datoteka *Microsoft* – prelazak u druga *Microsoft* aplikacija.

E sad, možete li zamisliti koliko tu ima ikona? Ne, kad posao jednom krene, deo po deo se brzo savlada, bilo uz pomoć odličnog Helpa, bilo uz pomoć još bolje dokumentacije.

U četvrtom redu iznad tabele je edit linija gde se vide podaci koje unosite pre nego što ih ubacite u ćeliju. Kada unosite ili korigujete podatke na početku reda imate 4 oznake. Strelica nadole može prikazati listu definisanih imena u tabeli, znak X prekida editovanje, znak kukice je zamena za Enter, a znak f() poziva *Function Wizard*. On je zanimljiv jer, kad ga aktivirate, daje spisak svih funkcija, vi odaberete onu koja vam treba i samo unesete potrebne argumente. Zgodno.

Tabele je na klasičan način označena slovima abecede za kolona i brojevima za redove. Ispod i sa desne strane tabela su Scroll Bar-ovi koji služe za kretanje po tabeli, odnosno prikazivanje njenih različitih delova. U dnu ekrana u zadnjem redu je status bar, gde se vidi objašnjenje za izabranu komandu.

To je ono što se vidi na ekranu, a treba odmah reći da je omogućen i komforan prelazak na *Excel za Windows* iz drugih spredšitova, kao što je recimo *Quattro Pro*, a pogotovu 1-2-3. *Excel za Windows* direktno učitava datoteke iz, recimo, 1-2-3 a može ih i snimiti u istom formatu (.WK*). Kompatibilnost je solidna, dosta bolje nego u prethodnim verzijama, a to se pogotovu odnosi na prilično korektnu konverziju Lotusovog WYSIWYG-a. Što se makro procedura tiče, preko Macro Translation Assistant a se one mogu prevesti i koristiti u Excelu.

U ovoj verziji Excela je izvršena reorganizacija strukture menija da bi se postigla veća kompatibilnost sa drugim MS aplikacijama, kao što je recimo, Microsoft Word. Korisnici prethodne verzije neće imati neprilika zbog toga, jer je jedan deo Helpa posvećen tome.

3D MODELI

U programu ima dosta novina, a mi ćemo se zadržati na jednom delu koji već u prvom momentu privlači pažnju.

S obzirom da se radi o trodimenzionalnoj tabeli, za pravljenje jakih takozvanih 3D modela odnosno aplikacija predviđeno je da se mogu kreirati 3D formule i funkcije, kao i 3D imena oblasti ćelija. Iz svih tabela se može automatski izvući grand total, što je za one koji dosta rade sa ciframa i te kako korisno.

Mada spredšit programi u principu nisu namenjeni za veći unos i obradu teksta, tom delu je u Excelu posvećeno dosta pažnje. Naime, moguće je poravnavanje teksta po želji (uz levu, desnu marginu ili centrirano) i to kako u jednoj tako i u više ćelija. Dalje, deo teksta ili samo jedan karakter se može drugačije naznačiti, istaći, a već standardne opcije su izbor različitih tipova i veličina fontove, te zadavanje raznih formata (zadebljana, zakošena slova, jednostruko ili dvostruko podvlačenje). Zanimljivo je da je programu dodat isti Spelling Checker kao u Microsoft Wordu – naravno, važi samo za engleski jezik. Jednom podašeni formati se mogu pomoću Format Painter tastera kopirati u druge ćelije ili objekte. Dopalo mi se što se na vrlo lak način mogu menjati boje elemenata u tabeli, koristiti već gotovi mnogobrojni oblici pozadine ili izabrati neki od gotovih hedera ili futera.

Prikazivanje podataka u grafičkom obliku, pravljenje sopstvenih slika ili korišćenje gotovih je priča za sebe.

Grafičke prezentacija podataka je vrlo značajan deo posla. Ne treba ni govoriti kolike su i kakva prednosti grafičke prezentacija podataka nad nepreglednim i glomaznim tabelarnim. Da je to tako potvrđuje i činjenica da su se programeri Excel-a dosta potrudili da omoguće maksimalni komfor u radu na tom delu programa, uz mogućnost izbora jednog od 14 tipova grafikona.

KREIRANJE GRAFIKONA

Za maksimalno pojednostavljenje rada služi *ChartWizard*, serija dijalog boksova kroz koje korak po korak kreirate grafikon profesionalnog izgleda. Za sve vreme rada na grafikonu vidite šta radite, možete vratiti korak nazad, pa tek na kraju, kada ste potpuno zadovoljni, završiti rad sa *ChartWizard*-om. Podrazumeva se da su podaci u grafikonu vezani za one u tabeli, te da se svaka promena vrednosti u tabeli automatski reflektuje na grafikon. Možete kreirati takve izveštaje u kojima će kao objekti, pored teksta i tabela, biti povezani i grafikoni. I u tom slučaju će svaka promena u originalnoj tabeli sa kojom su objekti povezani izazvati automatsku promenu u izveštaju koji ćete, recimo, štampati. S obzirom da **book** ima više tabela, možete u više njih kreirati grafikone koji prikazuju različite podatke, ili iste podatke na različite načine. To je korisno ako pripremate neku prezentaciju vašeg rada. Već se podrazumeva da svaki ili gotovo svaki element na grafikonu možete dodatno popraviti, prepraviti, dodati nešto novo. Korisna stvar je i mogućnost promene veličine grafikona u pripremi za štampanje. Imate tri opcije:

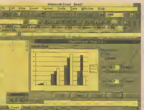
- **Full Page**, gde se veličina grafikona prilagođava veličini papira na kome štampate, a da pri tome ostavlja margine.
- **Scale To Fit Page** – grafikon zadržava proporcije kao što ga vidite na ekranu, ali popunjava celu stranu papira na kome štampate.
- **Custom**, gde je veličina grafikona nezavisna od veličine strane, onako kako ga vidite na ekranu, dakle u istim proporcijama i istoj poziciji. Pomoću ove opcije možete pozicionirati grafikon bilo gde na stranici papiru na kome štampate.

Interesantno je, a već viđeno u *Quattro Pro*, da grafikonu možete dodati takozvanu liniju trenda koja će pokazati trend ili pravac kretanja podataka koje prikazujete. Najčešće se koristi u regresionoj analizi, a ima pet tipova regresionih trend linija: linearne, logaritmičke, polinomialne i eksponencialne.

Čitav posao je, ma kako bio važan i ozbiljan, veoma lagan, čak zabavan, a sam *ChartWizard* je urađen fantastično, mnogo bolje nego u prethodnoj verziji.

GRAFIČKE VARIJACIJE

Tu nije kraj priče o grafici u *Excel*-u. Na primer, možete ubaciti grafički objekat u tabelu, grafikon ili dijalog tabelu. Grafički objekat nije stvarni deo tabele ili grafikona, već je ubačen preko njih i može se pomerati, povećavati, smanjivati ili formatirati nezavisno od podataka „ispod". Crtanje grafičkog objekta ide aktiviranjem **Drawing** ikone, a zatim preko dugmića u **Drawing** tulbaru, pri čemu morate imati miša. Kao što rekoh, **Drawing** tulbar ima 20-tak svojih ikona čijim aktiviranjem radite određene poslove: crtanje linije, strelica, slobodnoručno crtanje linija, pravougaonika, uglova, krugova i elipsi. Sve se to može bojiti, oblikovati i popunjavati po želji. Možete otvoriti i ubaciti tekst-boks u koji ćete upisivati tekst i formatirati ga kako vam odgovara. Taj tekst možete povezati sa tekstom u ćeliji. Grafičke objekte možete postaviti tako da se delimično preklapaju, možete birati koji će biti iznad koga, odnosno kako će se preklapati, a možete i više objekata grupisati i kasnije tretirati kao jedan. Dakle, u mogućnosti ste da pravite prave slike u svojoj tabeli, ali i da ih kopirate iz drugih aplikacija. Ja sam probao i uspeo da ubacim par slika iz raznih programa (.BMP, .PCX, .CGM) i to bez ikakvih problema. Pri tom je vrlo zanimljivo rešen postupak, jer pre ubacivanja slike u dijalog boksu možete da birate datoteke i da ih vidite, pa tek onda ube-



ChartWizard **korisniku nudi uzorak grafikona**



Veliki izbor tipova grafikona

cite preko tabele. Ubačena slika može se povećavati, smanjivati, pomerati, farbati... Odlično pa i više od toga.

U *Excel*-u 5.0 podaci se mogu organizovati kroz **List**, koji predstavlja seriju redova sa sličnim sadržajem. Na primer, **List** može biti lista klijenata sa adresama i telefonskim brojevima ili baza predračuna.

List može biti iskorišćen kao **baza podataka**, gde su redovi rekordi, a kolone polja (*field*). U okviru toga moguće su sve klasične spredšit opcije, kao što su sortiranje, pretraživanje, privremeno skrivanje podataka koji ne zadovoljavaju zadate kriterijume, a zgodna novost je automatsko dodavanje subtotala i totala. Podatke iz liste možete sortirati po abecedi, numerički ili hronološki, zavisno od sadržaja određene kolone ili kolona. Za prikazivanje, editovanje i brisanje pojedinih rekorda koristićete dijalog boks koji se zove **Data Form**. Tu je i **AutoFilter** uz koji možete jednim klikom naći kolonu koja sadrži odabranu stavku. Čak možete korišćenjem **Advanced Filter**-a kreirati svoj kriterijum za pretraživanje koji može biti prilično kompleksan. Sve je to na kraju začinjeno lepim mogućnostima za kreiranje odgovarajućeg izveštaja u koji, naravno, možete ubaciti i grafiku i posle svega ga štampati.

Kao zasebna celina u *Excel* je dodat i *Microsoft Query* koji služi za unošenje podataka iz drugih baza i učitavanje podataka iz drugih *Excel for Windows* „knjiga" (*Workbooks*).

Excel ima drajvere za preuzimanje podataka iz sledećih aplikacija: *Microsoft Fox Pro* 2.0 i 2.5, *Microsoft Access* 1.0 i 1.1, *Paradox* 3.0 i 3.5, *dBASE* 3.0 i 4.0, *SQL Server* 1.1, 4.2 i NT i *Sunbase 4.x*, *Microsoft Excel 3.0, 4.0, 5.0*, *Btrieve 5.1*, *ORACLE Server 6.0*, *ODBC ODS Gateway*, *Text*.

Osim što učitava podatke, *Query* može da ih preuzme iz više baza, a prema zadatim kriterijumima da odabere fildove koje hoćete da prikažete, da prikaže, edituje i organizuje rezultate pre nego što se vratite u *Excel* da izračuna totale za rekorde kroz **Result Set**. Kada se vratite u *Excel* podaci će biti u formatu ćelija *Excel*-a, a ne kao u *MS Query*. Stari korisnici *Excel*-a koji su radili sa O+E lako će konvertovati svoje makroe za ovu novu verziju.

RAD NA SCENARIJU

I prethodnu verziju *Excel*-a su odlikovale mnogobrojne mogućnosti za analizu podataka. Jedna od novosti u ovoj verziji je kreiranje novog *Scenario Manager*-a. O čemu se tu radi? U *Excel*-u 5.0 kao scenarije možete kreirati i sačuvati setove ulaznih vrednosti koji prouzrokuju različite rezultate. Scenario je grupa ulaznih vrednosti nazvanih „promenljive ćelije" koje su sačuvane pod nekim vašim imenom. Svaki set promenljivih ćelija reprezentuje set „šta ako" pretpostavki koje možete primeniti na worksheet model da biste videli efekte na druge delove modela. Možete definisati najviše 32 promenljive ćelije u jednom scenariju.

Scenario Manager možete koristiti za kreiranje višestrukih scenarija sa višestrukim setovima promenljivih ćelija, pregled rezultata svakog scenarija u svojoj tabeli, kreiranje sumarnog izveštaja sa svim ulaznim vrednostima i rezultatima, objedinjenje više scenarija u jedan model, zaštitu scenarija od modifikacija ili skrivanja, automatsko praćenje modifikacija.

Za rad sa scenarijom koristi se **Scenario** boks u **Workgroup** tulbaru. Pomoću njega se može brzo definisati i prikazati scenario u tabeli. Takođe **Scenario** komanda iz **Tools** menija omogućuju dodatne opcije u radu na scenariju.

Za analizu problema sa više promenljivih tu je i dalje *Solver* kao snažna alatka za optimizaciju. Koristi se kada hoćete da nađete optimalnu vrednost u određenoj ćeliji korigovanjem vrednosti u nekoliko ćelija ili kada hoćete da primenite specifična limite na jednu ili više vrednosti koje učestvuju u kalkulaciji. Inače *Solver* može analizirati tri tipa problema: linearne, nelinearne i celobrojne. S obzirom da je *Solver* dodatni program on se posebno instalira kroz *Microsoft Excel Setup* program.

Nije zaobiđena ni statistička analiza podataka. Za to služi specijalni set alata za analizu koji se zove *Analysis ToolPak*. Ti alati uključuju statističku analizu koju možete primeniti na više tipova podataka, a posebno su tretirane analize specijalno dizajnirane za inženjerske aplikacije (na primer *Fourier Analysis*).

Što se tiče podešavanja same tabele, odnosno njenog prilagođavanja korisniku, i tu ima novih rešenja. Jedno od korisnijih je mogućnost da se istovremeno otvori više *Workbook*-ova, radnih tabela, pa da se sve to zajedno snimi kao jedna datoteka nazvana **Workspace** (radni prostor). *Workspace* nije u stvari klasična *worksheet* datoteka i ne sadrži u sebi sve otvorene tabele, već je to datoteka sa informacijama o tome koji *workbook* je otvoren, gde se nalazi, njegova veličina i pozicija na ekranu. Kada otvorite Workspace datoteku, otvorene su i pozicionirane sve *workbook* datoteke onako kako ste ih imali pre snimanja.

U podešavanju tabele nudi vam se i mogućnost da sebi prilagodite tulbarove tako što ćete ubaciti ona ikone koje najčešće koristite. Možete napraviti i svoj novi tulbar sa sopstvenim ikonama koje će odrađivati posao za koji ste ih predvideli, tj. programirali. Sliku na ikoni možete napraviti ili prepraviti, doterati, ulepšati staru i to vrlo jednostavno. Osim toga tulbarove ili delove možete postavljati u samu tabelu na mesto gde vam to odgovara, ili ih „parkirati" uz levu, desnu ili donju ivicu ispod tabele.

PROGRAMIRANJE UZ VISUAL BASIC

Već sam spomenuo da je jedna od najvećih novosti to da u *Excel*-u možete programirati u *Visual Basic*-u. Naime, sve instrukcije kao delovi makroa su pisane u *Visual Basic*-u, pravom programskom jeziku. Dakle, u *Excel*-u možete pokrenuti *Visual Basic Programming System Application Edition* i automatizovati poslove u tabeli te doterivati aplikaciju po želji i mogućnostima odnosno znanju. Na žalost, iako smatram da je ovo velika stvar i napredak u programu ovakvog tipa, nisam programer i ne poznajem *Visual Basic* da bih mogao da vam dam više informacija. No, oni koji budu radili sa ovom verzijom *Excel*-a kad tad će se odlučiti da se okušaju i na ovom polju jer to daleko pomera limite spredšit programa. Kreatori *Excel*-a su, predviđajući da dosta korisnika ne zna *Visual Basic*, omogućili kao prvi korak snimanje korisnikovih izdavanja komandi pomoću takozvanog *Macro Recorder*-a. On će za nas napisati makro. Nakon nekog vremena kada korisnik nauči da snima i pokreće makroe verovatno će poželeti da ih poboljša dodavanjem sopstvenog *Visual Basic* koda, a kažu da je *Visual Basic* lak za učenje.

Nemoguće je u ovako kratkom predstavljanju spomenuti svo mnoštvo mogućnosti koje *Excel* za *Windows* verzija 5.0 nudi. Sve u svemu, nakon prvog iskustva u radu sa ovim programom mogu mirne duše da kažem da opravdano nosi titulu najboljeg na svetu u svojoj oblasti. Nadam se da ćemo uskoro imati priliku da vidimo šta se dešava u konkurentskom taboru – mislim na Lotus – ali prosto ne mogu da verujem da se može napraviti bolji program za radne tabele.

Korisna adresa
MISOFT 11000 Beograd, Skadarska 45 Tel. 011 343-043


PC plus, Budimska 10 a, (011) 337-120, 340-227
... AND YOU'LL NEVER WORK ALONE...

KANCELARIJA BEZ PAPIRA

Računarski sistemi za izradu, obradu, organizaciju i arhiviranje dokumentacije (CDMS) na personalnim računarima danas predstavljaju zrela i proverena rešenja, koja i malim i velikim firmama mogu pomoći da svoje računare povežu u efikasan informacioni sistem i zauvek se reše „papirologije".

Vladan Aleksić

Pijuckajući svoju prvu jutarnju kafu, gospodin Petrović je polako prelistavao dnevnu poštu na monitoru računara na svom stolu. Faks poruke primljene preko mrežnog faks modema pregledao je direktno sa mreže, a pošta pristigla tog jutra odmah je skenirana u prijemnoj kancelariji i takođe prosleđena kroz mrežu. Listovi su se polako smenjivali na ekranu. Bilo je tu zahteva, dopisa, ponuda, najnoviji Službeni list, prospekti, poruke pristigle elektronskom poštom... Na pojedinim listovima g. Petrović bi upisivao svoje komentare, zaokruživao bitna mesta, precrtavao neprihvatljivo, a zatim ih upućivao kroz mrežu na dalju obradu nadležnom službeniku. Pre toga bi svakome od njih „zavirio" u „ulaznu korpu" sa dokumentima koji čekaju na obradu, pa bi na osnovu toga odabrao onoga koji je najmanje opterećen poslom.

Naredni list na ekranu je bio ponovljeni zahtev važnog kupca za ponudom. „Pa ovo sam još prošle sedmice dao da se radi", promrmlja ljutito i upisa ključne reči u masku procedure za pronalaženje dokumenta u obradi: zahtev, ponuda, ime kupca. Već narednog trenutka program odgovori imenom i prezimenom službenika. „Opet Lazić!" gotovo uzviknu i odabra strelicom na ekranu polje sa imenom. Pred njim se otvori dugačak spisak neobrađenih dokumenata kod preopterećenog službenika. *Lazić da se odmah javi kod šefa!* prosledi poruku kroz mrežu, pa krenu dalje kroz pristiglu poštu. Pažnju mu privuče šareni prospekt jarkih boja. Uveliča sliku u boji, pa sa pomenom na tabeli sa tehničkim podacima. Uređaj mu se učini neverovatno sličan nečemu što se upravo projektuje u tehničkoj službi. U polje maska za pretraživanje arhive tehničke dokumentacije upisa naziv projekta, pa iz ponuđenog spiska odabra naziv crteža, koji se već narednog trenutka pojavi na ekranu. Uveliča detalj, pa pređe na sliku sa prospekta, pa ponovo na crtež...

Na ovo ću se još vratiti, pomisli pre nego što je nastavio da lista poštu. A onda mu se lice ozari. „Konačno!" pomisli ugledavši list sa slikom koja mu je bila potrebna da bi kompletirao cirkularno pismo koje je čekalo da bude poslato već nekoliko dana. Otvori program za obradu teksta, učita već ranije pripremljeni tekst i u ostavljen prazan okvir umetnu sliku koju je maločas našao. „Gotovo!" reče zadovoljno, pozva masku za kontrolu mrežnog faksa i jednim potezom miša obeleži trideset imena sa ponuđenog spiska. Jedno „klik" i mrežni faks server je preuzeo dalju brigu o dopisu: popuniće automatski ime i adresu u dopisu za svakog od primalaca i poslati ga na odgovarajući broj čim to telefonska linija dozvoli.

"Dobro jutro, šefe", začu se bojažljivi glas istovremeno sa kucanjem. Kroz odškrinuta vrata kancelarije provirivao je zbunjeni Lazić.

GDE JE PROBLEM?

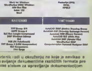
O poslovnom sistemu bez papira, u kojem se svi podaci čuvaju u bazama podataka, a svi poslovi obavljaju pomoću računara, počelo je više da se govori pre petnaestak godina, uporedo sa pojavom naprednijih programa za obradu teksta, tabelarne proračune. Negde u isto vreme počela je da se otima i ideja o projektnom birou bez table, olovke i lenjira, zajedno sa prvim CAD programima za personalne računare. Priče umnogome slične onoj s početka ovog teksta su ponovljene mnogo puta kao primamljive propagandne vizije softverskih kuća. Pojedinačni programi su u međuvremenu razvijeni do neslućenih razmera. Tehnologija rada u okviru pojedinačnih poslova se neprestano naglo razvijala, a mogućnosti, kvalitet i brzina rada vrtoglavo rasli. Prvo smo obrađivali samo tekst, zatim smo ga kombinovali sa automatskim tabelama, pa smo onda počeli da u njega ubacujemo i crteže i slike. Prvo u crno-beloj tehnici, a zatim i u boji. Prvo smo samo crtali tehničke crteže, zatim smo počeli da gradimo prostorne mrežne modele, a na kraju ih vizuelizovali i animirali sa visokim stepenom realizma. Na žalost, sa svakom novom verzijom bilo kojeg od programa, početna vizija o „bespapirnoj sredini" se bukvalno gušila u brdima papira koje izbacuju sve brži i kvalitetniji štampači i ploteri. Umesto da papir nestaje ili da se bar potiskuje, njegove su se količine beznadežno povećavale. Papir je i dalje ostao osnovno sredstvo za komunikaciju među ljudima u procesu rada, a razvijale se samo tehnologije za pripremu i obradu njegovog sadržaja. Ma kako to paradoksalno izgledalo, ovakvo stanje je bilo uslovljeno pre svega realnim ograničenjima poslovnog okruženja.

Informacije u digitalnom računarskom obliku po svojoj prirodi svakako ne zahtevaju papir kao nosioca. Da bi ona, međutim, bila dostupna korisniku na mestu gde mu je potrebna, on mora imati računar koji uz to mora imati odgovarajuću komunikaciju sa računarom gde se informacija čuva, kao i program koji je u stanju da je pročita i prikaže s obzirom na oblik u kome je sačuvana. Sve te zahteve je u uslovima realnog poslovnog okruženja i korišćenja standardnih gotovih programa vrlo teško ispuniti. Najlakši način da se premosti taj jaz je opet papir. Ako informaciju prenesemo na papir, možemo je poneti bilo gde i može je koristiti bilo ko bez ikakvih posebnih tehničkih uslova. Tako se krug zatvara. Osim toga, jedna realna poslovna celina mora da stalno komunicira sa okruženjem. Komunikacije putem papira i dalje je dominantna, a i kada se one ostvaruju računarskim sredstvima, i pored izvesne standardizacije ostaje ozbiljan problem međusobnog prepoznavanja različitih formata računarskog zapisa.

DELOVI I CELINA

Ako je donedavno osnovno težište u razvoju poslovnog i tehničkog softvera bio razvoj programa koji unapređuju funkcije i povećavaju performanse u obavljanju konkretnih poslova u određenoj oblasti, onda je to danas nesumnjivo povezivanje standardnih pojedinačnih programa i njihova integracija u kompleksne, fleksibilne i univerzalne informacione sisteme. Na taj način korisnik nastavlja da radi u poznatom okruženju standardnih programa za pojedine namene na koje je navikao, a upravljanje dokumentima različitih formata koji u njima nastaju preuzima sistem za upravljanje dokumentacijom. Ovi sistemi su poznati pod opštim nazivom, a njihov podsistem za upravljanje tehničkom dokumentacijom se najčešće nazivaju EDM (*Engineering Document Management*).

Začeci ovakvih sistema su programska rešenja za arhiviranje različitih vrsta informacija, takozvane „multimedijalne" baze podataka. Njihov prvobitni zadatak je bio da organizuju racionalno čuvanje tekstova, tabela, slike, tehničkih crteža i modela formiranih pomoću računara. Bez obzira na međusobnu različitost, sve ove oblike opisuje i kontroliše ista baza podataka. Cilj je da se dokumentima pristupa na jedinstven način i da se kroz iste maske omogući njihovo pretraživanje, pregled pa i štampa, bez obzira na sasvim različito poreklo i karakter. Naime, za korisnika je potpuno nevažno u kakvom se računarskom obliku nalazi neki dokument. Njegov je zahtev uvek da srodna i funkcionalno povezana dokumenta budu povezane i kroz jedinstvenu bazu podataka i da se do njih dolazi na jedinstven i najjednostavniji mogući način. Ako su za jedan predmet vezani slike u boji (PCX, TIF, GIF ili TGA), tehnički crtež u vektorskom – CAD obliku (DWG ili DXF), skenirani crtež (RLC ili Tiff CCITT Grupa 4) i tekstualni opis (DOC ili TXT), korisnik želi da svim ovim dokumentima brzo i lako priđe i da ih pogleda ili štampa, koristeći samo jedan program i je



Korisnik radi u okruženju na koje je navikao, a upravljanje dokumentima različitih formata preuzima sistem za upravljanje dokumentacijom



CDMS – Corporate Document Management System

dinstven kriterijum za pretraživanje. Da bi podaci uopšte mogli da se prenesu do svakog od potencijalnih korisnika i održavaju na jedinstven način, računari u okviru poslovnog sistema moraju da budu povezani u mrežu. Jedno od najpoznatijih rešenja ove vrste za PC računare, koje je već steklo punu afirmaciju i na našem tržištu, je sistem programa RX EDM, proizvod norveške firme RASTEREX.

RX SPOTLIGHT

U toku skeniranja redovno se javlja jedan broj grešaka, odnosno reprodukuju se sve nepravilnosti sa originala. Za njihovo ispravljanje, odnosno retuširanje skeniranog crteža koristi se *RX Spotlight*. Program prihvata crteže svih formata i rezolucija skeniranja, i čini ga univerzalno primenljivim bez obzira na struku i vrstu dokumenata. Podržani formati su RLC (*Run Length Coded*), RLP (kompresovan RLC), TIFF (nekompresovan, CCITT grupe 3 & 4, Bit Pack), CALS, Windows BMP i Z Soft PCX, u monohromatskom zapisu.

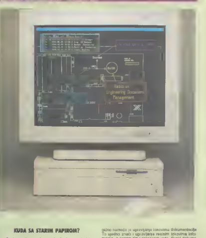
Nakon učitavanja rasterskog crteža na ekranu, prvo se otklanjaju grube greške. Jedna od tipičnih je ispravljanje zakošenja crteže koje nastaje uglavnom usled lošeg postavljanja papira u skener. Karakteristične tačke koje definišu pravac iskrivljenja se definišu prvo grubo, a zatim na uveličanom detalju i precizno. Na osnovu ove linije *Spotlight* poravnava celu rastersku sliku. Druga tipična greška je pojava „snega", odnosno odraz hrapavosti papira ili sitnih nečistoća na njegovoj površini. Pored ovih komandi, na raspolaganju stoje i automatski filteri čija primena može umnogome poboljšati izgled rasterske slike i time olakšati eventualnu kasniju vektorizaciju. Od posebnog značaja je i mogućnost selekcije oblasti na koju želimo da primenimo izabrani filter. Program podržava sledeće filtere:

- *median*: „zaglađivanje" konture uklanjanjem ili dodavanjem tačaka po ivicama
- *coupling contour*: spajanje razdvojenih u jedinstvene konture
- *inversion*: invertuje piksele u odabranoj oblasti crteža
- *thining*: istanjuje linije
- *thickening*: podebljava linije
- *shape enhancer*: popravlja konturu rasterskog objekta uklanjajući nepotrebne piksele
- *line separator*: vrši razdvajanje delimično spojenih rasterskih linija

Pored opcija filtriranja, *Spotlight* nudi i skup funkcije rasterskog crtanja. Na raspolaganju su slobodnoručno crtanje, kao i definisanje pravilnih geometrijskih oblika. Tekst koristi standardne SHX fontove koje program pronađe u AutoCAD direktorijumu. Svi novi grafički objekti betiraju se vektorski do trenutka kada se crtež snimi na disk, tako da se slobodno mogu pomerati i modifikovati.

Ono što je posebno interesantno u *Spotlight*-u su mogućnosti manipulacije rasterskim objektima komandama na koje smo do sada uglavnom nailazili u vektorski orijentisanim programima. Rasterske elemente je moguće odabrati različitim kriterijumima (pomoću prozora ili proizvoljne izlomljene linije), a zatim ih kopirati, brisati, pomerati, zakretati ili ih uvećavati i umanjivati. Program dozvoljava da se do devet ovako odabranih delova rasterskog crteža privremeno sačuva kao „simboli" za kasniju upotrebu.

U radu sa postojećom tehničkom dokumentacijom značajna je i mogućnost uvođenja vektorskih crteža u postojeću rastersku sliku. Šta to konkretno znači vidi se na sledećem primeru. Na raspolaganju nam je crtež urađen rukom na pausu, na kojem je hitno potrebno izvršiti zamenu nekog elementa. U AutoCAD-u se nacrta željeni element i snimi kao poseban crtež. Postojeći crtež se skenira i uvede u *Spotlight*, a zatim se nekom od funkcija za brisanje ukloni prethodni i napravi prostor za novi elemenat. Preostaje još samo uvođenje vektorskog elementa, koji se dovede na željenu poziciju u crtežu i aplicira. Kao rezultat imamo rasterski crtež sa potrebnom izmenom, koji se zatim može brzo odštampati i proslediti u proizvodnju.



KUDA SA STARIM PAPIROM?

Svako realno poslovno okruženje pored dokumentacije nastale na računaru poseduje i dokumentaciju nastalu tradicionalnim sredstvima. Ona se čuva na papiru, a deo nje se i dalje stalno ili povremeno koristi. Čuvanje određene dokumentacije u nekom vremenskom periodu predstavlja i zakonsku obavezu. Direktno prevođenje u računarski oblik čitljiv za standardne programe za obradu dokumenata (obrada teksta ili CAD) nije realno, suviše je komplikovano i dugotrajno, što znači i vrlo skupo. Umesto toga, ovaj deo dokumentacije se prevodi u digitalni oblik skeniranjem, a u računarskoj arhivi se čuva kao slika dokumenta.

Po potrebi, bilo koji od dokumenata se može i prevesti u oblik pogodan za izmene na računaru. U slučaju tekstova za to se koriste programi za automatsko prepoznavanje teksta (OCR – *Optical Character Recognition*), dok se tehnički crteži vektorizuju digitalizacijom po ekranu (*RX AutoIcon*) ili automatskom vektorizacijom odgovarajućim programima (*RX Vectory*). Ako se obezbedi da se i celokupna dokumentacija koja stiže iz okruženja (pošta, prospekti, časopisi, obrasci, formulari...) svakodnevno redovno skenira i stavlja pod kontrolu informacionog sistema, prvi put se zaista stiču uslovi da se u jednoj ograničenoj poslovnoj celini celokupan tok dokumentacije i interne komunikacije odvija kroz informacioni sistem, a bez posredstva papira. Papir se pojavljuje kao neizbežno sredstvo komunikacije samo na periferiji takvog sistema: na ulazu u „prijemnoj kancelariji", gde se celokupan ulaz skenira, i na izlazu, gde se konačno uobličena dokumenta nastala u sistemu štampaju i šalju partnerima u okruženju.

UPRAVLJANJE TOKOVIMA DOKUMENTACIJE

Konačno, kada se celokupna dokumentacija u procesu nalazi u računarskom sistemu, cilj koji se logično nameće je upravljanje tokovima dokumentacije. To ujedno znači i upravljanje realnim tokovima informacija, a samim tim i procesom rada. Svaki dokumenat u takvom sistemu prati procedura definisana na ulazu ili na mestu nastanka dokumenta i određuje kome u sistemu, kojim redosledom i sa kojim ovlašćenjima se dokumenat prosleđuje. Ova procedura može biti standardna ili se formira prema konkretnom dokumentu. U svakom trenutku se može utvrditi na kom mestu (kod kog učesnika u procesu) se dokumenat nalazi i kakav je njegov trenutni status (u radu, gotov – neoveren, overan, arhiviran...). Svakom od učesnika u procesu rada sistem formira takozvanu ulaznu i izlaznu „korpu" (*in basket* i *out basket*), gde automatski ili na nečiji poseban zahtev smešta dokumenta za koje je on nadležan. Na taj način je onemogućeno da se neki dokumenat zagubi ili da ga korisnik zanemari. Potpuno je nemoguće i da se utvrđene procedura ne ispoštuje do kraja. Ako i pored svega dođe do greške, vrlo je jednostavno ustanoviti ko je za nju u sistemu odgovoran. O definisanju i izvođenju procedura za upravljanje dokumentacijom brine se program *Work flow*.

S obzirom na svoje mogućnosti, sistemi za upravljanje dokumentacijom su logičan, gotovo neizbežan izbor prilikom uvođenja sistema kvaliteta (evropski standard kvaliteta ISO 9000). Sistem kvaliteta se naravno može uvesti i bez upotrebe računara, pa i sistema za upravljanje dokumentacijom, ali je sigurno da je tek uz njihovu primenu zaista efikasan.

TEHNIČKA DOKUMENTACIJA U SISTEMU

U okviru opštih sistema za upravljanje dokumentacijom, upravljanje tehničkom dokumentacijom (EDM) zaslužuje posebnu pažnju kao specifičan podsistem. Održavanje i organizacija tehničke dokumentacije je obično mnogo veći problem nego njena proizvodnja. Svaki crtež u toku stvaranja prolazi kroz više faza, konačno rešenje mora uvek da se umnoža-

RX VECTORY

Jedan broj crteža ipak je neophodno prevesti u vektorski oblik, pre svega zbog većeg obima izmena. RX VECTORY omogućava prevođenje još uvek aktuelne papirne dokumentacije u vektorski oblik čitljiv za CAD programe, a pre svega AutoCAD.

RX Vectory je jedan od retkih programa za vektorizaciju koji prikazuje polazni rasterski crtež na ekranu, omogućava izmene na njemu i prikazuje tok vektorizacije. Nakon uvođenja rasterskog crteža u program na njemu je moguće vršiti razne izmene. Na raspolaganju su filteri slični onima u programu *RX Spotlight*. Po završetku obrade rasterske slike definišu se parametri vektorizacije. Program predlaže nekoliko skupova predhodno definisanih parametara vektorizacije, posebno prilagođenih vrsti crteža koji vektorizuje: mašinski, geodetski, arhitektonski, elektrotehnički ili crtež iz neke druge oblasti. Naravno, svaki od parametara se može definisati i nezavisno od predloženih vrednosti. Najznačajniji parametri su:

- *fit polyline* definiše preciznost za aproksimaciju krivih linija
- *dimension* određuje veličinu najmanjeg objekta koji će biti vektorizovan
- *gap jump* definiše dužinu najvećeg prekida u rasterskom objektu koji se premošćava prilikom vektorizacije
- *snap to ortho* eliminiše male odstupanja od horizontale i vertikale
- *text as* način prepoznavanja teksta

Oblast nad kojom se vrši proces vektorizacije može biti ceo rasterski crtež ali i bilo koji njegov deo. Isto tako moguće je definisati i masku oblasti nad kojom se neće vršiti vektorizacija. Postupak se može pokrenuti i iz „batch moda" tako da korisnik zada niz crteža koje treba sukcesivno vektorizovati.

Sama vektorizacija nije prosto provlačenje linija kroz rasterske konture. *Vectory* prepoznaje entitete kao što su linija, luk i krug i kao takve ih prevodi. Isto tako, krug ukršten dvema linijama prevešća identično u vektorski oblik, a ne kao 8 linija i 4 luka. Tekst može biti pretvoran u linijski oblik, prepoznat (*Optical Character Recognition* metodom) ili potpuno uklonjen sa crteža. Kote se prevode u AutoCAD-ove parametarske kote i čuvaju na posebnom sloju crteža. Rezultat vektorizacije se prikazuju preklopljeni preko rasterske slike (uz mogućnost njenog isključenja) tako da se odmah po završetku posla mogu uporediti sa originalnim crtežom. Vektorski crtež se može sačuvati u formatima DXF, DXB, VCT ili direktno u *AutoCAD*-ovom DWG formatu zapisa.

RX HIGHLIGHT

Jedan od najvažnijih modula EDM sistema je i *RX Highlight* koji omogućuje brz i lak pregled arhiviranih dokumenata bez obzira na format zapisa i način njihovog nastanka. Pomoću njega dokumenti se mogu pregledati, komentarisati, uređivati i štampati. Program podržava veoma veliki broj formata zapisa, a spisak podržanih se stalno uvećava. Od vektorskih formata crteža do sada su podržani sledeći: AutoCAD DWG, DXF, DXB, SLD i RND, MicroStation DGN, Personal Designer DRW, Hewlett Packard HPGL i HPGL2. Osim toga program omogućuje i rad sa sledećim rasterskim monohromatskim i kolor formatima zapisa: TIF, RLC, GIF, TGA, BMP, RLE, PCX, DIB, LBM, IFF, WPG, kao i tekstualnim dokumentima nastalim pomoću *Word*-a, *WordPerfect*-a ili bilo kog ASCII editora teksta i tabelama formiranim pomoću *Excel*-a i *Lotus*-a.

U ovom programu nije moguće vršiti nikakve izmene na originalnom dokumentu, već samo postavljanje različitih stepena uvećanja i pregled po nivoima (u slučaju vektorskih zapisa). Osim toga, program sadrži i kompletan set komandi za crtanje i pisanje teksta kao komentara. Ovako nacrtani sadržaj se ne može uvesti u originalni dokument, već uvek ostaje na posebnom nivou koji se snima i vezuje uz njega. U okviru komentara se čuvaju podaci o datumu i vremenu njegovog nastanka i imenu korisnika koji ga je formirao. Uz dokument se može sačuvati više različitih komentara, a svaki se čuva na posebnom nivou i u posebnoj boji, uz informaciju o autoru i vremenu nastanka.

Formiranje komentara postaje posebno značajna opcija u mrežnim sistemima jer se pomoću njih u okviru mreže razmenjuju sugestije i poruke o dokumentu među pojedinim stručnjacima koji sistem koriste. Osim toga, sličicama sačuvanim na nivou komentara mogu se predložiti izmene na dokumentima koje će kao nova revizija u dokument biti unesene tek kasnije.

Mogućnost poređenja vektorskih i monohromatskih rasterskih crteža preklapanjem na ekranu obezbeđuje brzo i jednostavno sagledavanje izmene nastalih prilikom njihovih revizija. Crteža koji se upoređuju program preklapa i različitim bojama obeležava delove koji se razlikuju.

RX INDEX

RX Index je program za kontrolu baze podataka o grafičkoj i tekstualnoj dokumentaciji, ali u isto vreme i svojevrsna „kontrolne table" sa koje se na jednostavan i pregledan način pozivaju i svi ostali programi u sistemu i na koju se korisnik automatski vraća nakon završetka rada u bilo kom od njih.

Pomoću ovog alata korisnik brzo i lako može definisati polja u bazi, izgled ekrana kao i raspored atributa pojedinih polja, i na taj način formirati aplikacije za organizaciju različitih arhiva poslovne i tehničke dokumentacije, kataloga eksponata ili proizvoda, baza podataka o licima i slično. Veza sa standardno korišćenim programima u sistemu (editori teksta, rasterskih i vektorskih crteža, program za pregled i komentarisanje dokumenata itd.) i ostale posebne funkcije se ostvaruju pomoću jednostavnog i efikasnog skript jezika koji *RX Index* interpretira.

Rad sa definisanom bazom se ostvaruje preko grafičkog interfejsa jednostavnim unošenjem poznatih kriterijuma u za to određeno polje. Pri tome program nudi efikasan sistem spiskova za izbor (*choice list*) pa se ništa od konkretnog sadržaja polja ne mora pamtiti, a pretraživanje postaje lako i efikasno. Pošto se definiše kriterijum, program pretraži bazu i prikaže spisak datoteka koje zadovoljavaju postavljeni kriterijum. Izgled spiska za izbor se takođe može definisati slobodno, a u okviru njega se može pojaviti bilo koja kombinacija podataka iz definisane baze. Krećući se po spisku, korisnik dobija umanjenu i pojednostavljenu sliku tekućeg dokumenta u prozoru. Kada se odabere jedan ili više dokumenata sa spiska, on se, u skladu sa ovlašćenjima korisnika, mogu pregledati, štampati, komentarisati ili menjati.

RX Index je prilagođen za rad u mreži. Do sada su podržani samo DOS orijentisani mrežni operativni sistemi kao što je *Novell NetWare*, a u razvoju su i verzije koje će podržavati i druge računarske platforme i operativne sisteme.

va u više primeraka, a na njemu se i naknadno unose izmene. U praksi krajnji ishod je uvek sličan: mnogo papira u opticaju, a na njima najrazličitije verzije crteža, i naravno uvek pogrešna kod ključnog čoveka. Činjenica da je firma otpočela sa uvođenjem konstrukcije ili projektovanja na računaru (CAD) obično situaciju sa dokumentacijom čini samo još složenijom. Tada se deo dokumentacije čuva na računaru, deo se i dalje proizvodi na klasičan način, a sva stara dokumentacija ostaje u klasičnoj arhivi. Da bi se ovakvo stanje prevazišlo, treba prevesti kompletnu dokumentaciju na računar bez obzira na vrstu, poreklo i oblik i organizovati bazu podataka koja je prati. Samo takva organizacija može da omogući da se do svakog dokumenta dođe brzo i efikasno, tražeći ga po bilo kom kriterijumu. A da bi sva dokumentacija bila dostupna svima kojima može biti potrebna istovremeno, pojedinačne računare treba povezati u mrežu.

Ovakva ideja je donedavno bila veoma teško izvodljiva, i to pre svega zbog raznorodnosti oblika podataka u zavisnosti od načina njihovog unosa. Nova dokumentacija rađena u CAD-u čuva se u vektorskom obliku (DWG datoteka iz AutoCAD-a), postojeća nakon skeniranja ostaje u rasterskom obliku (RLC datoteke koje opisuju matricu tačaka iz kojih su sastavljena linije na crtežu), a tabelarni podaci o crtežima se upisuju u neku od standardnih baza podataka (na primer DBF). Za manipulaciju pojedinim vrstama podataka odavno su na raspolaganju pojedinačni programi različitih proizvođača, koje je najčešće bilo teško efikasno i pouzdano povezati. Ako je to i postignuto u jednom trenutku, pojava nova verzije bilo kog od segmenata najčešće se pretvarala u noćnu moru. Celovito rešenje koje uspešno povezuje sva poznata standardna programa u ovoj oblasti je Rasterexova linija programa za organizaciju, izradu, održavanje i arhiviranje tehničke dokumentacije RX EDM (*Engineering Drawing Management*), podržana procesorskim grafičkim karticama Laser AT i Tracer. Poštujući dva već uspostavljena faktička standarda, AutoCAD kao vektorski program za proizvodnju CAD dokumentacije i format baze podataka DBF, Rasterex je dodao sve što je nedostajalo i povezao sve delove u jedinstven sistem, uključujući i specijalizovana periferijske uređaje kao što su grafička table, ploteri i skeneri.

RX EDM nudi rešenja za sve nivoe potrebe korisnika. S obzirom na to da programi za pretraživanje baze podataka o crtežima (RX Index) i pregled i štampu crteža (RX Highlight) rade i na svakoj VGA kompatibilnoj grafičkoj karti i pod MS Windows 3.1, RX EDM dostupan i korisnicima sa skromnijom opremom. Uz njihovu pomoć korisnik AutoCAD-a dobija pravu priliku da organizuje svoj rad: nema više razmišljanja o rogobatnim skraćenim nazivima crteža i zamornog traganja po direktorijumima. RX Index ga udobno vodi kroz bazu podataka u kojoj je jedno od polja i umanjena i uprošćena slika crteža. Ako želi da samo pregleda crteža ili da ih štampa, ne mora ni da ulazi u AutoCAD i čeka na dugotrajno učitavanje crteža. RX Index će ga odvesti u RX Highlight gde se crtež pojavljuje mnogo brže i gde može da ga uveličava na ekranu ili štampa. Ako mu brzina grafičkih operacija na ekranu nije dovoljna (što je najčešće slučaj), na raspolaganju su mu Rasterexove procesorske grafičke kartice čija je grafička brzina u *AutoCAD*-u i ostalim pomenutim programima izuzetna.

Veliki korisnici koji imaju potrebe da i svoju staru dokumentaciju prebace u grafičkom obliku u računar za dalje korišćenje u RX EDM u će takođe naći sva potrebna rešenja. *RX Spotlight* će omogućiti da se efikasno otklone sve greške nastale tokom skeniranja, *RX RasterCAD* dozvoljava selektivnu, a *RX Vectory* vektorizaciju celog crteža. I sve to opet i na brzim Rasterexovim karticama.

Da bi se kompletni sistemi mogli zaokružiti, kod svakog korisnika potrebno je obezbediti i odgovarajuće periferijska uređaje: ink-jet, termalne ili LED (elektrofotografske) plotere i skenere odgovarajućih formata, kao i optičke diskove ili „juke-box" uređaje dovoljno velikih kapaciteta za smeštaj velike količine tehničke dokumentacije.

UMESTO ZAKLJUČKA

Računarski sistemi za izradu, obradu, organizaciju i arhiviranje dokumentacije (CDMS) na personalnim računarima danas predstavljaju zrela i proverena rešenja koja daju ostvarljiva rezultate uz umerena ulaganja. Oni uspešno povezuju i integrišu postojeće afirmisane programe za obradu teksta, digitalnih tabela, crno-belih i slika u boji i tehničkih crteža prevazilazeći razlike u formatima zapisa među njima. CDMS omogućava da se organizuju multimedijalne baze podataka, tako da je istim programima i pomoću istih kriterijuma moguće lako i brzo pronaći, pregledati ili štampati bilo koji traženi dokument bez obzira na njegov format zapisa. S obzirom na svoju fleksibilnost i modularnost, ovi sistemi su podjednako primenljivi i za mala i za najveće poslovne sisteme u najrazličitijim oblastima rada. Po svojoj prirodi zahtevaju da pojedinačni računari budu povezani u mrežu, a obezbeđuju i izuzetno važnu funkciju upravljanja tokovima dokumentacije prethodno utvrđenim procedurama. Ostvarujući maksimalno svoju funkciju povezivanja i integracije, CDMS istinski pretvara ne pojedinačnih računara u efikasan informacioni sistem.

Korisna adresa

OSA
11000 Beograd, Narodnog fronta 56
Tel. 011 681 199, 644-567

R2-D2, PC KONTROLER

Prvi put Vam prikazujemo naš novi proizvod pod nazivom R2-D2. To je industrijski sistem za akviziciju podataka, merenje i procesno upravljanje. On je istovremeno kompletan PC sistem, veoma snažan PLC, ili podstanica distribuiranog sistema (DCS). R2-D2 podržava PC kompatibilnost, DOS, WINDOWS, UNIX, QNX, operativne sisteme, PCMCIA i standardne diskove ako su upotrebljeni. Takođe, mogu se instalirati RAM-ROM disk emulatori (LABTECH u EPROM'u) za primene u okruženjima gde upotreba standardnih diskova nije moguća. R2-D2 je potpuno otvoren sistem i kompatibilan je sa bilo kojim PC hardverskim dodacima kao i sa popularnim softverskim alatima.



Za R2-D2 postoji veliki broj analognih i digitalnih ulazno-izlaznih modula.R2-D2 se može konfigurisati do 224kanala. Moduli su kompatibilni sa serijom modula PCI-20000 od kompanije BURR-BROWN I INELLIGENT INSTRUMENTATION za koje postoji veliki broj softverskih paketa za akviziciju podataka, merenje i procesno upravljanje.

R2-D2 je napravljen da zameni sličnu opremu od kompanija SIEMENS, OMRON, FESTO ALEN-BRADLEY itd, kao i od domaćih proizvodjača, ali za višestruko nižu cenu, često viših performansi i sa potpunom PC kompatibilnošću. Na primer:

R2-D2, 386-40 MHz, kao na slici sa RAM-ROM diskom (0K) košta samo **995***

ELEKTRONIK DIZAJN ima ovako nisku cenu prvenstveno zbog svojih orginalnih tehničkih rešenja (uskoro patent). Danas, <u>ni jedan</u> proizvodjač slične opreme u svetu <u>ne može</u> ponuditi ništa slično za ovu sumu. Garantni rok je 3 godine, servisni rok je 7 godina i tehnička dokumentacija je na srpskom jeziku a isporuka je u roku od 2 nedelje! Specifikacije IP54 ili MIL-810D su na raspolaganju na Vaš zahtev.

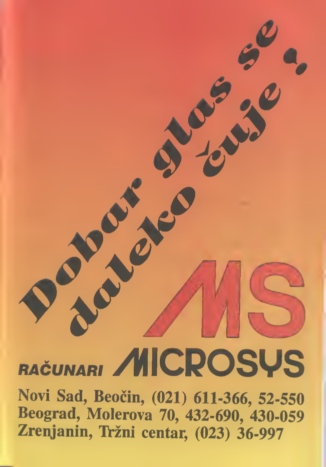
Posetite nas na SAJMU TEHNIKE u hali 1, štand 1313. Pokazaćemo Vam naš novi proizvodni program. ELEKTRONIK DIZAJN ima i nove za 30% niže cene sistema za akviziciju podataka, merenje i procesno upravljanje. Za više informacija pozovite 011/450-480, radnim danom od 9 do 19h. Fax: 011/444-74-59

Electronic Design

DVE GODINE ISPRED

Maršala Tolbuhina b.b.-Pejton, Beograd

***) Bodova. Ulazno-izlazni moduli nisu uključeni u cenu.**


Dobar glas se
daleko čuje
MS
RAČUNARI MICROSYS
Novi Sad, Beočin, (021) 611-366, 52-550
Beograd, Molerova 70, 432-690, 430-059
Zrenjanin, Tržni centar, (023) 36-997

PROJEKTOVANJE PROGRAMA

Vreme pisanja programa od nekoliko stotina linija definitivno prolazi: sve se više traže gigantski programi koji predstavljaju kompletne informacione sisteme i koje razvijaju čitavi timovi po nekoliko godina. U skladu sa tim, menjaju se i alatke potrebne za pisanje programa. *Artist*, nastao na beogradskom FON-u, je „pravi" CASE alat, koji pruža pomoć u najvažnijoj fazi rada – projektovanju programa.

Nenad Bartolomin

Dok je nekad za pisanje programa bilo dovoljno imati editor, prevodilac i dobru volju, sada je za produktivan rad neophodno imati mnogo više. Savremeni prevodioci se prodaju sa obiljem dodatnih programa koji olakšavaju i ubrzavaju pisanje programa. U takve programe ulaze razni generatori izveštaja, dizajneri ekrana i maski za unos, generatori koda i sl. U suštini sve više se radi na automatizaciji pisanja programa. Naravno, još smo svetlosnim godinama udaljeni od „programa koji će sami pisati programe" – programiranje je još uvek visoko kreativna delatnost koja se ne može „simulirati" veštačkim sredstvima. Međutim, ono što prosečan programer obično radi sadrži i značajan deo manje više fizičkog posla: dizajniranje ekrana i izveštaja, programiranje klasičnih situacija i sl. su tipični primeri. Osnovna ideja je da se dragocen i kreativan rad programera što više oslobodi od nevažnih detalja, tako se smanjuje cena programa i mogućnost greške u automatski generisanim procedurama.

Osim pomenutih alatki koje olakšavaju sam proces programiranja, već dosta dugo postoje i alati koji pomažu programeru i u drugim fazama razvoja programa. To su takozvani *CASE* alati (*computer-aided software engineering*). Na žalost, ovi programi se kod nas relativno retko koriste i to po pravilu na velikim projektima koji imaju sredstava za nabavku ovih prilično skupih programa. Međutim, neophodnost i prednosti korišćenja *CASE* programa je učinila da se na tržištu pojavi dosta proizvoda koji su sasvim prihvatljivi po ceni i za one sa manjim sredstvima. Sve je više ovakvih programa koji se distribuiraju čak i kao shareware/PD proizvodi. Ono što oni nude se ne može uvek meriti sa visoko profesionalnim proizvodima, ali svakako može biti korisno u nekim fazama pisanja programa.

KAKO SE PIŠE PROGRAM?

Pisanje programa je u suštini veoma složen proces i deli se u nekoliko faza. Grubo gledano, dijagram tog procesa se može predstaviti jednim grafom (slika 1) koji počinje **projektovanjem**, nastavlja se **pisanjem programa**, **testiranjem** i **korišćenjem**, a zatim se krug zatvara. Faze testiranja i korišćenja skoro uvek imaju dva izlaza: jedan je kraj rada na programu i ovaj izlaz se (ma kako to čudno zvučalo) veoma retko koristi. Mnogo češće korišćena grana vodi ponovo u fazu projektovanja ili (ako imate sreće) u fazu pisanja programa. Ovakav životni ciklus programa je neminovan i svako ko je pisao malo duže programe sve kako će ga uočiti bez ikakvog „teorijskog" predznanja. Naravno, svaka faza ovog ciklusa može proces vratiti i jedan korak nazad, u prethodnu fazu.

Ono što je veoma bitno u ovom ciklusu (a često se stavlja u drugi plan) je faza projektovanja. Programeri veoma često zanemaruju ovu prvu fazu i odmah po dogovorenom poslu počinju da ukucavaju program u računar. Ovaj pristup donekle ima svoje opravdanje u našim ekonomskim prilikama: retko koji naručilac će prihvatiti da uloži novac u nešto što smatra „nekorisnim" i što direktno ne rezultuje opipljivim rezultatima! Na žalost, posledice ovakvog pristupa su veoma brzo vidljive: loše projektovan sistem je veoma nestabilan i teško se menja, javljaju se nedostaci koje je veoma teško a ponekad i nemoguće otkloniti itd.

Treba znati da se prve faze u stvari i ne može preskočiti: one se u opisanom slučaju jednostavno spaja sa pisanjem programa, pa se projektovanje obavlja **zajedno** sa pisanjem programa. Naravno, ukoliko imate sreće i iskustva, i ovako se može napisati sasvim upotrebljiv program, ali su izgledi da nešto zapne veoma povećani.

Uloga *CASE* alata je da programeru pomognu u svim fazama razvoja složenog programskog sistema. U zavisnosti od toga koje se faze automatizuje, na tržištu se može naći veoma raznolika ponuda. Zbog široke definicije, u *CASE* alate se mogu ubrojati i sasvim prosečne pomoćne alatke koje umereno pomažu programeru. Obično ovakvi programi pomažu u dizajniranju ekrana i sl., što spada u fazu pisanja programa. Međutim, „pravi" *CASE* alati obavezno pomažu u prvoj fazi: projektovanju programa.

PROJEKTOVANJE PROGRAMA

Ako se preskoče stroge definicije, projektovanje programa se obično svodi na planiranje podataka koji će biti korišćeni u programu i funkcije koje program treba da obavi. Ovo nema veći značaj za manje programe, ali za veće sisteme (primera radi, za informacioni sistem jedne firme) je apsolutno neophodno. Način na koji se ovo radi može biti različit: u krajnjoj liniji možete olovkom na papiru skicirati osnovne odnose u nekoj svojoj notaciji. Međutim, daleko je bolje to učiniti na neki standardan način. Metod koji se najčešće koristi u *CASE* programima zove se **strukturna analiza sistema**. Pod sistemom se ovde podrazumeva sistem koji se „preslikava" u programski sistem – znači realni sistem koji modeliramo programom.

Realni sistem se pri korišćenju ove metode predstavlja **dijagramom toka podataka**. Na tom dijagramu se vide tokovi i skladišta podataka, osnovni procesi i veze između njih. Na slici 2. je prikazan jedan takav dijagram. **Skladišta podataka** predstavljaju podatke: to može biti jedna ili više tabela u različitoj relacionoj bazi podataka. Skladišta podataka se obeležavaju nazivom između dve paralelne linije. Na primer, skladište podataka na slici 2. je objekat **studenti**.

Elipsama (ili krugovima) se predstavljaju procesi. Na primer, **evidencija studenata** je jedan od procesa u datom primeru. Procesi mogu biti različite složenosti – ako se radi o jednostavnijim procesima koji se dalje ne razlažu, onda se procesu osim imena dodeljuje i kratak opis njegovog funkcionisanja ili algoritam rada u nekom pseudo jeziku.

Pravougaonici predstavljaju tzv. **terminatore**. To su objekti koji mogu biti van posmatranog sistema, ali su važni kao izvori ili primaoci nekih tokova podataka. Na primer, na slici 2. objekat **student** je jedan od terminatora.

Svi objekti u dijagramu se povezuju numerisanim linijama koje predstavljaju **tokove podataka**. Tokovi podataka, na primer, mogu povezivati proces i terminator kako bi se označila njihova komunikacija. Na primer, veza **Studenti** spaja terminator **Studentska služba** i proces **Evidencija studenata**.

Opisanim načinom se funkcije sistema i njegova struktura predstavljaju na veoma pregledan i lako shvatljiv način. U principu, ovaj metod nije vezan za primenu računara i može se bez problema „ručno" koristiti. Čak za crtanje ovakvih dijagrama mogu se koristiti i standardni programi za crtanje. Međutim, realni problemi zahtevaju veoma česte izmene i daleko su složeniji od datog primera. Tada *CASE* program pokazuje svoju veliku prednost: ako pomerite jedan objekat, sve veze koje su vodile do njega takođe se ažuriraju, za šta bi bilo potrebno dosta truda na papiru ili u nekom programu za crtanje. Međutim, i to nije preveliki problem. Ovakvi dijagrami mogu biti veoma kompleksni i bilo bi nemoguće celu šemu sa detaljima pregledno smestiti na jedan crtež. Zato se primenjuje **hijerarhijska dekompozicija**: osnovni dijagram sadrži samo najvažnije elemente koji se dalje razlažu na nove dijagrame. Na primer, svaki složeni proces u dijagramu može se razložiti na novi dijagram u kome se sada pojavljuju detalji: novi procesi sadržani u polaznom, skladišta podataka, terminatori i tokovi. Tako se olakšava shvatanje celine koju svi dijagrami zajedno predstavljaju.

Sledeća važna uloga *CASE* alata je u proveri kreiranih dijagrama: svaki dijagram toka podataka treba da ispunjava određene uslove, počev od jednostavnih uslova tipa „svaki čvor dijagrama treba da bude povezan sa nekim drugim" do veoma složenih međuzavisnosti koje se zadaju rečnikom podataka. Naime, pri crtanju dijagrama se za svaki njegov element (proces, vezu...) zadaje opis koji se upisuje u rečnik podataka. Informacije koje se upisuju u rečnik mogu biti različite (zavisi od konkretne implementacije), ali se obično upisuje naziv elementa, tekstualni opis i, u zavisnosti od tipa (proces, veza ili nešto treće) upisuju se algoritam rada u pseudo jeziku, struktura podataka i sl. Na primer, kada se zadaje tok podataka **izdata_knjiga** treba zadati strukturu koja opisuje taj tok. Na primer, to može biti:

```
izdata_knjiga    čitalac +
                 knjiga +
                 datum_izdavanja
```

Ovim se zadaje da će se tok podataka u bazi podataka predstavljati tabelom koja sadrži tri kolone: identifikacije čitaoca, identifikacija knjige i datum izdavanja. Ove informacije se kasnije koriste pri proveri integriteta rečnika i generisanju dokumentacije.

ARTIST

Artist je *CASE* alat projektovan i urađen u Laboratoriji za informacione sisteme Fakulteta organizacionih nauka iz Beograda. Urađen je kao standardni *Windows* program sa hardverskim zahtevima koji donekle potiču iz zahteva samih *Windows*-a: računar mora biti najmanje 386 sa 4MB RAM memorije. Ove konfiguracije je, na žalost, zaista minimum, jer će *Artist* na njoj neke operacije raditi veoma sporo, a složenije operacije nikako... za normalan rad treba imati bar 8 MB RAM-a. *Artist* na disku zauzima nešto malo više od 1 MB, što je zaista skromno. Program koristi sve pogodnosti *Windows* okruženja, što naročito dolazi do izražaja kada nešto treba odštampati: možete koristiti standardne *Windows* fontove i instalirane drajvere za štampač. Naravno, kompletna komunikacija sa korisnikom se odvija preko poznatih *Windows* menija i maski za unos.

U prvoj fazi projektovanja informacionog sistema, *Artist* koristi standardnu strukturnu analizu sistema. Prvo treba kreirati tzv. **kontekstni dijagram**: to je standardni dijagram toka podataka ali na najvišem nivou. Može sadržati samo jedan proces – ovaj dijagram je predviđen kao najgrublji i najopštiji model realnog sistema ili jednog njegovog dela. Na tom modelu se vide ulazi i izlazi iz sistema, kao i odnos posmatranog sistema i okoline.

Sledeća faza je preciznije modeliranje sistema: sada se svaki proces „razgrađuje" kako bi se dobio precizniji opis. U suštini, svaki proces se dalje dekomponuje u novi dijagram toka. Interesantno je (i veoma korisno) da *Artist* omogućava automatsko crtanje dijagrama nižeg nivoa: kada se aktivira opcija za dekompoziciju procesa, nudi se kreiranje dijagrama „od nule" ili automatsko kreiranje. Naravno, automatsko kreiranje ne može zameniti projektanta i dijagram koji se kreira može biti samo kostur koji se dalje razvija, ali i to je značajno olakšanje. Jer, svaki dijagram-naslednik mora da zadrži „okolinu" procesa: ako je proces koji se dekomponuje imao izlaz „izdavanje_knjiga", onda i dijagram naslednik mora da ima takav izlaz. Na primer, dijagram sa slike 2. je nastao dekompozicijom procesa sa slike 4.

Samo kreiranje dijagrama je veoma jednostavno

Slika 1

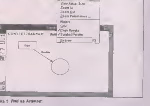


Faze razvoja programa

Slika 2



Primer dijagrama toka podataka



Slika 3 Rad sa Artistom

Slika 6

```
skladište podataka   Studenti
    {
        Podaci o studentu
    }
    tokovi podataka koji ažuriraju ovo skladište podataka
        Studenti u procesu 2 Evidencija studenata
    tokovi podataka koji pretražuju ovo skladište podataka
        Studenti u procesu 4 Formiranje izveštaja o testu
skladište podataka   Predmeti i testovi
    {
        <
            šifra predmeta     ID ključ
            naziv predmeta     Naziv
            {
                Test
            }
        >
    }
    tokovi podataka koji ažuriraju ovo skladište podataka
        Predmeti i testovi u procesu 1 Evidencija predmeta
    tokovi podataka koji pretražuju ovo skladište podataka
        Predmeti i testovi u procesu 4 Formiranje izveštaja o testu
        Predmeti i testovi u procesu 3 Obrada rezultata testa
skladište podataka   Rezultati testa
    <
        šifra predmeta     ID ključ
        naziv predmeta     Naziv
        rb_testa    broj
        {
            broj indeksa     broj
            ime studenta     Naziv
            ocena     broj
        }
    >
    tokovi podataka koji ažuriraju ovo skladište podataka
        Rezultati testa u procesu 3 Obrada rezultata testa
    tokovi podataka koji pretražuju ovo skladište podataka
        Rezultati testa u procesu 4 Formiranje izveštaja o testu
```

Izveštaj o objektima u sistemskom rečniku

iz menija ili iz palete ikona izabere se odgovarajući objekat, izabere se mesto na ekranu i jedan klik na taster miša postavlja izabrani objekat na željeno mesto. Posle ovoga *Artist* traži da se unesu ostali podaci o kreiranom objektu. Veze se takođe jednostavno kreiraju: prvo se izabere odgovarajuća ikonica, označi se objekat koji predstavlja izvor veze, zatim objekat koji je kraj veze. Slično se mogu menjati i veze između objekata, zakrivljenost i sl. Pomeranje i promena dimenzije objekata je urađena na način uobičajen za programe za crtanje: selektuje se objekat i povlačenjem odgovarajuće kontrolne tačke menja se dimenzija/proporcija objekta. Pomeranje je takođe jednostavno: označi se objekat koji se pomera i držeći taster miša „odnese se" na drugo mesto. Veoma je važno da se posle svake od ovih operacija ažurira ceo dijagram: sve veze koje vode do objekta koji se pomera automatski se ažuriraju, pa je slika uvek korektna.

Podaci o svim elementima dijagrama upisuju se u rečnik sistema. Dok se elementi dijagrama u rečniku automatski menjaju kako se menja slika, neki delovi se moraju menjati direktno u rečniku. Taj deo rečnika je standardni rečnik podataka. *Artist* između ostalog u rečniku čuva podatke o **domenima, strukturama** i **korisničkim funkcijama**. Za objekte iz svake od ovih kategorija postoji mogućnost da se direktno upisuju u rečnik, menjaju ili brišu korišćenjem odgovarajućih opcija.

Domeni su u stvari tipovi podataka. Naziv „domen" se standardno koristi jer polja i elementi strukture uzimaju vrednosti iz određenih domena. Kada se kreira nov domen, zadaje se njegova osnovna pripadnost (niz simbola, ceo broj, realni broj, datum, logička vrednost), naziv i opis. Može se zadati i način na koji je domen implementiran u konkretnom slučaju: na primer, ne moraju svi sistemi imati tip datum, za to se može sasvim lepo koristiti i niz simbola. Opciono se može zadati i ograničenje na vrednost. Treba zadati logički izraz koji proverava da li vrednost ulazi u specificirani domen. Pri zadavanju ovog izraza mogu se koristiti standardni logički operatori **not**, **or**, **and** kao i **betwen** za proveru gornje i donje granice, **in** za pripadnost skupu, **substring** za izdvajanje dela stringa. Kao deo izraza mogu se upotrebiti i funkcije koje će kasnije biti definisane. Ove funkcije se takođe mogu upisivati kao posebni objekti u rečnik, s tim što se osim naziva upisuje još samo opis funkcije.

Osnovni tipovi podataka (domeni) se koriste prilikom definisanja **polja** – to su obično buduće kolone u tabelama baze podataka. Polja se dalje koriste da bi se definisale **strukture**, njima se opisuju tokovi podataka i terminatori (u terminologiji *Artista* – **interfejsi**), ali se mogu koristiti i samostalno za istu svrhu. Podržana su četiri osnovna tipa strukture: agregacija, ekskluzivna, neekskluzivna unija i skup. Agregacija je „najobičnija" struktura: njeni elementi su polja ili neke druge strukture koje je čine. Na primer, struktura **čitalac** može se definisati na sledeći način:

```
čitalac
<
    ime         string,
    prezime     string,
    adresa      string,
    dat_rodj    datum
    id_br       id
>
```

Ako struktura opisuje objekat koji uzima samo jedan tip vrednosti od nekoliko izabranih, koristi se ekskluzivna unija koja se označava srednjim zagradama.

```
metod plaćanja
[
    gotovina    broj,
    ček         dokumenta,
    kartica     dokumenta
]
```

Ukoliko objekat uzima jednu ili više vrednosti iz ponuđenog skupa, koristi se neekskluzivna unija:

```
pod o studentu
/
    stud.dom    dokumenta,
    kredit      dokumenta
/
```

Objekat se može opisivati i skupom. Na primer, svaka faktura može imati više stavki:

```
stavka fakture
{
    artikal     string,
    količina    broj,
    cena        broj
}
```

Prethodni zapis opisuje stavke fakture kao **skup** čiji je svaki element predstavljen trojkom (**artikal, količina, cena**).

Elementi strukture ne moraju biti samo osnovni tipovi koji se definišu u rečniku podataka, to takođe mogu biti strukture. Na primer

```
student
<
    ime        string
    prezime    string
    rodjen     datum
    ispiti
      {
        naziv       string,
        ocena       broj
        ispitivač   profesor
      }
>
```

Definisanje ovakvih struktura je veoma jednostavno prilikom kreiranja (ili crtanja) objekta korisnik unosi odgovarajuću strukturu koja opisuje objekat. Pri tom može koristiti prethodno definisane domene i strukture kao domene za elemente nove strukture. Jedina mala teškoća je što ne postoji mogućnost da se neki domen naknadno definiše za vreme kreiranja strukture ako se u pola posla setite da vam treba domen „datum" moraćete da prekinete unos podataka za tekući objekat, startujete odgovarajuću opciju menija, unesete u rečnik novi domen i tek onda nastavite sa radom. Međutim, ukoliko pri definisanju strukture želite da definišete novu strukturu koja će vam trebati samo na tom mestu, nije potrebno posebno definisanje, jednostavno se umesto domena navede definicija strukture i Artist će je automatski smestiti u rečnik kao naimenovanu strukturu. Na primer, takve definicije je polje **ispiti** iz prethodnog primera.

Korisničke funkcije se upotrebljavaju prilikom zadavanja raznih ograničenja domena. Na primer, matični broj građana je niz cifara, ali mora zadovoljavati određene uslove: početni niz označava datum rođenja, ceo niz ima kontrolnu cifru itd. Takav uslov se proverava funkcijom koja sužava ceo skup svih nizova cifara na nizove cifara koji zadovoljavaju uslove za matične brojeve. Korisničke funkcije se takođe čuvaju u rečniku sistema. Kao što je već rečeno, za njih se zadaje samo neformalni opis i naziv funkcije.

Artist prilikom rada ne dozvoljava korisniku da naruši pravila integriteta u sistemskom rečniku i sprečava svaku takvu akciju. Celokupan rad je organizovan preko tzv. **transakcija**. Naime, akcije korisnika se ne upisuju odmah u rečnik sistema, već se sve obavlja u memoriji. Tek kada se obavi skup operacija koji predstavlja (po korisniku i posle provere pravila integriteta) jednu zaokruženu celinu, transakcija se završava i nastale izmene se upisuju u rečnik sistema. *Artist* pamti stanje transakcija čak i ako prekinete rad sa programom, sledeće startovanje će vas vratiti tačno tamo gde ste stali. U toku rada je moguće definisati „kontrolnu" tačku na koju se program vraća po izboru odgovarajuće opcije – veoma zgodno za kraće eksperimente u radu. *Artist* ima i klasične **Undo/Redo** operacije koje omogućavaju da se precizno poništava dejstvo obavljenih operacija; ukoliko želite da poništite dejstvo poslednje operacije, izabraćete operaciju **Undo**. Operacija **Redo** poništava dejstvo **Undo** operacije, tj. ponovo vraća stari efekat.

SPECIFIKACIJA BAZE PODATAKA

Strukturna analiza sistema služi za opisivanje i analizu celog sistema. Pri tom se podaci u sistemu u dijagramima prikazuju preko tokova i skladišta podataka. Iz tih dijagrama se može videti koji je odnos procesa i podataka i koji proces koristi koje podatke. Da bi se posao kompletirao, neophodno je specificirati i samu bazu podataka, znači podatke u sistemu i njihove međusobne odnose. Za to postoji više metoda, a najpopularnija je metoda **objekti-veze**, o kojoj je već bilo reči u „Računarima". *Artist* koristi modifikovanu varijantu ove metode (prošireni model objekti-veze). Ukratko, radi se o načinu za prikazivanje strukture baze podataka, njenih elemenata i uzajamnih veza. Osnovni objekti se prikazuju pravougaonicima (u terminologiji *Artista* **kernel**). Kernel može biti bilo koji realni objekat (odnosno klasa objekata) koji je od interesa za informacioni sistem koji pravimo, na primer **kupci**, **fakture**... Objekti se međusobno povezuju **relacijama** (vezama) koje se prikazuju romboidima. Na primer, u bazi podataka za biblioteku vezom **napisao** mogu se spojiti objekti **pisac** i **knjiga**. Svaki objekat i veze mogu imati više atributa koji predstavlja neko svojstvo objekta. U stvari, atributi izdvajaju jedan objekat iz cele klase, na primer, objekat **knjiga** predstavlja celu klasu knjiga u biblioteci, atributi **naziv**, **izdavač** i sl. izdvajaju jednu knjigu iz čitave klase. Atributi se predstavljaju elipsama.

Slika 4



Kontekstni dijagram

Slika 5

```
REM ORACLE database table definitions dbschem

REM drop table          Student

create table          Student (
        Index NUMBER (6) NOT NULL
        ImeStud CHAR (32)
   primary key (
        Index
   )
)

REM drop table          Polaganje;

create table          Polaganje (

   primary key (

   )
)

REM drop table          Predmet

create table          Predmet (
        SifPred CHAR (32) NOT NULL
        NazPred CHAR (32),
   primary key (
        SifPred
   )
)

REM drop table          Polozio;

create table          Polozio (
        RBP NUMBER (6  NOT NULL
        VrstaPol CHAR (32)
        DatumPol DATE
        Ocena NUMBER (6)
   primary key (
        RBP
   )
)

REM drop table          Test

create table          Test (
        SifPred CHAR (32) NOT NULL
        Sifratesta NUMBER (6) NOT NULL
        Datumtesta DATE
        Maksimalnibrojpoena NUMBER (6)
        Tiptesta CHAR (1) NOT NULL
   primary key (
        SifPred
        Sifratesta

REM drop table          RezTest

create table          RezTest (
        Osvojenipoeni NUMBER (6)
   primary key
```

Generisana šema baze podataka

Osim ovih klasičnih elemenata *Artist* podržava i dodatne objekte: kernel može biti i **slab objekat** ukoliko njegovo postojanje zavisi od drugog objekta (odnosno od veza sa tim objektom). Na primer, postojanje stavki jedne fakture ima smisla samo ako postoji faktura. **Agregacija** je objekat koje je istovremeno i objekat i veza između dva objekta. Na primer, veza između pisaca i knjiga može biti objekat **napisao** – ovo je u suštini veza, ali predstavlja i samostalni objekat sa atributima **datum**, **rukopis** i sl. **Podtip** je objekat koji predstavlja neku užu klasu već postojećeg objekta. Na primer, podtip svih poslovnih partnera su kupci.

Cela baza podataka se može predstaviti skupom dijagrama objekti-veze. *Artist* automatski taj skup objedinjuje u jedan model čitave baze podataka. Korisnik prilikom kreiranja novog dijagrama ima mogućnost da iz rečnika podataka uzima već definisane objekte i tako pravi model čitave baze. Samo crtanje dijagrama je jednostavno i radi se slično kao i crtanje ostalih dijagrama: izabere se opcija **New Submodel** i tako se otvara nov prozor u kome se crta dijagram. Paleta ikona se sada proširuje ikonama za elemente dijagrama objekti-veze. Jednostavnim izborom odgovarajuće ikone (ili izborom odgovarajuće stavke menija) i pozicioniranjem na ekranu kreira se nov objekat u dijagramu. *Artist* zahteva prilikom svake ovakve operacije da se zadaju svi podaci za kreirani objekat: na primer, ako se kreira kernel, treba zadati naziv objekta (to može biti neki naziv koji je već u rečniku), opis i njegove atribute. Ukoliko se atribut prvi put kreira, unose se i podaci o njemu: naziv, opis, domen, da li je ključ. Osim ovoga, za objekat se mogu zadati i podaci o učestalosti koji su korisni kada se kasnije model implementira. Naravno, moguća je i kasnije izmena svih objekata dijagrama. *Artist* pritom ne dozvoljava nikakvu akciju koja bi narušila pravila integriteta.

Veza sa dijagramima toka podataka ostvaruje se u toku kreiranja ovih dijagrama, odnosno zadavanja skladišta i tokova podataka: za svako skladište i tok može se zadati spisak dijagrama objekti-veze na kojima je predstavljen objekat.

IZVEŠTAJI

Smisao *CASE* alata nije samo u crtanju dijagrama koji zadovoljavaju određene uslove kada završite sve dijagrame koji opisuju jedan informacioni sistem. *Artist* za vas može uraditi još dosta posla. Prvo, sve eventualne izmene će se automatski preneti na sve definisane dijagrame. Sledeća važna stvar je pisanje dokumentacije: ako ste prilikom zadavanja objekata u dijagramima i u sistemskom rečniku bili dosledni, *Artist* može dnı *Artist* može generisati veoma obimnu i kompletnu dokumentaciju. Ova dokumentacija sadrži podatke o svim objektima definisanim u sistemskom rečniku: procesi, skladišta, tokovi, terminatori, korisnička funkcije, strukture, domeni, elementi dijagrama objekti-veza itd. Ovde dolazi do izražaja prava moć centralizovane obrade svih podataka o sistemu: izveštaj će osim podataka koji su eksplicitno uneti za svaki objekat iz rečnika izvaditi i ostale podatke koji se odnose na njega. Generisana dokumentacija je dosta obimna i pokriva veći deo onoga što je potrebno za detaljnu specifikaciju jednog informacionog sistema.

Na osnovu podataka u rečniku može se generisati i **šema baze podataka**, radi se o konkretnoj implementaciji objekata iz rečnika. Pošto se u rečniku nalaze sva potrebna informacije o podacima, ovaj posao je potpuno automatski. *Artist* generiše kompletne podatke za *Oracle*, mada pretpostavljam da nije neki problem urediti da se dobijaju podaci i za druge sisteme. Generisana specifikacija sadrži sve što je potrebno za kreiranje svih tabela i indeksa u bazi (slika 5).

Osim šeme moguće je generisati izveštaj o svim ostalim elementima sadržanim u sistemskom rečniku. Na slici 6 je prikazan primer takvog izveštaja.

U praktičnom radu *Artist* se pokazuje kao veoma kvalitetno urađen program, samo *Windows* okruženje mu daje izgled visoko profesionalne aplikacije sa konzistentnim i jednostavnim korisničkim interfejsom. Posao koji obavlja je u suštini dosta komplikovan, pa relativno visoki hardverski zahtevi nisu neuobičajeni. Uz program se dobije dosta kratko (i veoma lepo složeno) uputstvo koje na žalost nije pisano previše popularno; čak i ako se pretpostavi predznanje o projektovanju informacionih sistema, uputstvo deluje dosta suvoparno i spartanski pisano. Ovo naročito dolazi do izražaja kad se uzme u obzir i da program nema uobičajan help sistem: u mnogim situacijama bi pritisak na help taster uštedeo zamorno prelistavanje dokumentacije. Međutim, i pored toga *Artist* zaslužuno zaslužuje pažnju: radi se o veoma dobrom programu koji je u potpunosti razvijen kod nas, zasnovanom na našim naučnim rezultatima. Razvoj ovog programa je još uvek aktuelan: dosadašnji razvoj je bio usmeren na razne oblike specifikacije informacionog sistema, a sada se radi na alatima koji će biti od pomoći u fazi implementacije informacionog sistema.

ADACOM

Čika Ljubina 12, BEOGRAD
Tel (011)629-233, 341-496 337-367
Fax. (011)629-233, 337-367

ADACOM - STANDARDNE KONFIGURACIJE

Konfiguracija	Opis	Cena
PC 486-66 VLB	AMD 4 MB RAM 270 MB HDD 5¼" i 3½" FDD SVGA color	**3,290**
PC 486-50 VLB	Intel, 4 MB RAM, 270 MB HDD, 5¼" i 3½" FDD, SVGA color	**3,290**
PC 486-40 VLB	AMD 4 MB RAM 270 MB HDD 5¼" i 3½" FDD SVGA color	**2,990**
PC 486-40 ISA	Intel, 4 MB RAM, 215 MB HDD, 5¼" FDD, SVGA mono	**2,390**
PC 386-40 DX	AMD 4 MB RAM 215 MB HDD 5¼" FDD SVGA mono	**1,810**
PC 386-40 SX	AMD, 2 MB RAM, 215 MB HDD, 5¼" FDD, SVGA mono	**1,530**

Doplatu računate kao razliku u ceni izabrane i standardne komponente

♦ OSTALO

Artikal	Cena
Printeri	
EPSON LX400 A4 9 pina	490
EPSON FX1170 A3 9 pina	1 200
EPSON LQ100, A4, 24 pina	590
EPSON LQ570 + A4 24 pina	880
EPSON LQ1070 + A3 24 pina	1 280
HP IVL, toner, 1 MB, 300x300	1 880
HP IV toner 2 MB 600x600	3 600
OLYMPIA AEG toner 1 5 MB 300x300	1,700
Besprekidno napajanje	
UPS MIKOM 500 VA/500 VA sa stabilizat.	470 / 520
UPS LONG TIME 300 VA	350
UPS LONG TIME 1000 VA	1 100
Fax-Modem	
ZOLTRIX MNP5 V 42 interni	170
DISCOVERY MNP5 V 42bis interni	250
ModemFax extern: DISCOVERY	pozovite
Dodaci	
Miš TAICHE	35
Miš TAICHE podloga adapter	60
Džojstik WARRIOR 5	50
Zvučna kartica AUDIO PLUS 8Bit	140
SOUND GALAXY NX PRO 8bit	300
SOUND GALAXY BASIC - 16bit	450
Ručni skener GENIUS 4500A, 256 gray	250
Filter za monitor stakleni/mrežasti	40 / 25
CENTRONIX kabl za printer	15
Kabl za napajanje 220V	10
Razno	
Koprocesor 80387 40MHz CYRIX	150
Strimer Interni COLORADO 120 / 250MB	450 / 550
Mrezna karta NE 2000 app 16 bit	150
Potrošna roba	
Diskete 3M/FUJI, 1 2 MB 5¼"	25
Diskete 3M/FUJI 1.44 MB 3½"	35
Diskete NoName 1 2 MB 5¼"	15
Diskete NoName, 1 44 MB 3½"	20
Strimer traka, 3M - 60 MB	45
Strimer traka 3M 120 MB	60
Riboni za štampač EPSON, A4	15
Riboni za štampač EPSON, A3	15

♦ KOMPONENTE

Komponenta	Cena	PC 386-40 SX	PC 386-40 DX	PC 486-40 ISA	PC 486-40 VLB	PC 486-50 VLB	PC 486-66 VLB	Vaš izbor
Osnovna ploča								
ISA 386SX 33 MHz SOYO-AMD	190	♦						
ISA 386DX-40 MHz 128K SOYO AMD	260		♦					
ISA 486DX-33 MHz 256kB SOYO INTEL	900			♦				
VLB 486DX 33 MHz 256kB SOYO-INTEL	950				♦			
VLB 486DX-50 MHz 256kB SOYO-INTEL	1,250					♦		
VLB 486DX2-66MHz 256kB SOYO-INTEL	1 400						♦	
Memorije								
1 MB / 70ns SIMM GOLD STAR	85							
2 MB / 70ns SIMM GOLD STAR	170	♦						
4 MB / 70ns SIMM GOLD STAR	340		♦	♦	♦	♦	♦	
Hard disk								
QUANTUM / IBM 170 MB / 14 ms	480	♦	♦	♦	♦	♦	♦	
QUANTUM / WD - 250 MB / 12 ms	550							
QUANTUM / MAXTOR - 340 MB / 12 ms	650							
SEAGATE ST3550A 460 MB / 11 ms	950							
CONNER CP3504 - 510 MB / 11 ms	1 150							
Flopi disk drajv								
5¼ 1 2 MB TEAC SAFRONIC	120	♦	♦	♦	♦	♦	♦	
3½" 1 44 MB TEAC SAFRONIC	100				♦	♦	♦	
Kontroler								
IDE AT bus 16 bit ASCA-UMC	40	♦	♦	♦				
IDE cache (4 x HDD + 2 x FDD)	320							
IDE AT bus 32 bit VLB VISION UMC	90				♦	♦	♦	
Video kartica								
UMC 800X600 256 kB	80							
TRIDENT T-9000 1024x768 512 kB	100	♦	♦	♦				
T-8900CL ET4000 CL5422 1 MB	170							
CIRRUS CL5428 1(2)MB 32bit VLB	230				♦	♦	♦	
PARADISE WD90C33 1(2)MB 32bit VLB	480							
Monitor								
14" SVGA-mono DTS 1024x768	280	♦	♦	♦				
14" SVGA color SONIGA 0 28 - NI	680				♦	♦	♦	
14" SVGA-color, AXIOM NI-LR 0 28	730							
Kućište								
DESK TOP PS 200W LEADMAN Display	140	♦	♦					
MINI TOWER PS 200 W A-TECH Disp	150			♦	♦	♦	♦	
Tastature								
102 tastera KB-102 ASCII GEAR	60	♦	♦	♦	♦	♦	♦	
101 taster YU/ASCII CHERRY	150							

EPSON, HEWLETT PACKARD CONNER QUANTUM SEAGATE, MAXTOR AEG TEAC PANASONIC GENIUS CREATIVE LABS FUJI NEC TOSHIBA TEXAS INSTRUMENTS CANNON, BORLAND DISCOVERY LONGSHINE, WESTERN DIGITAL, AT&T CYRIX IIT AMD INTEL, TRIDENT TSENG, ATI CIRRUS LOGIC PARADISE, IBM ADACOM 3M DISCOVERY MITSUMI SONY MICROSOFT

MATEMATIKA I NJENI SIMBOLI

Paket *Mathematica* u sebi sadrži programski jezik koji sa svojom elegantnom sintaksom, moćnim operacijama i bogatim bibliotekama raznih funkcija omogućava da i vrlo komplikovana izračunavanja ili definicije budu zapisani u svega nekoliko redova, a ono što najviše oduševljava jeste ideja potpuno simboličkog tretiranja svakog izraza. Po svemu osobena, *Mathematica* raspolaže i sa stotinama ugrađenih i eksternih operacija koje povećavaju njenu ionako veliku moć.

VIKTOR CEROVSKI

O programskom paketu Stivena Volframa *For doing mathematics by computer* bilo je reči na stranicama „Računara" u dve navrate (broj 75, str. 36 i broj 91, str. 40). U međuvremenu, objavljena je i prva knjiga domaćih autora o čuvenom programu. Biće potrebno još mnogo toga da se napiše pre nego što se stekne kompletna slika o mogućnostima *Matematike*. U ovom tekstu će biti reči o osobenostima *Matematikinog* programskog jezika.

Preporučena konfiguracija za korišćenje *Matematike* je računar sa 68020 procesorom i 32 Mb RAM-a. Program je međutim moguće koristiti i na nekoj slabijoj konfiguraciji – na primer PC-ju sa 486 procesorom i 8 Mb RAM-a ili čak običnom 386 računaru. Preciznije, postoje posebni programi za 386 konfiguraciju sa i bez koprocesora. Za smeštanje programa na hard-disk potrebno je oko 20Mb. Postoji takođe mogućnost instalacije programa za 386/387 i na računaru bez koprocesora – mnoge vrlo upotrebljive operacije (pretežno „simboličke" prirode) rade bez problema. S druge strane, računanje najobičnijeg 1+1 može da blokira računar. Ovo nimalo ne iznenađuje ako imamo na umu da koprocesor služi „samo" za brzo izvršavanje nekoliko osnovnih računskih operacija i elementarnih funkcija u jednostrukoj i dvostrukoj tačnosti.

Sam program ima nekoliko verzija, a najniža sa kojom je potpisnik ovog teksta radio nosi oznaku 1.2. Verzija 2.0 je donela niz unapređenja: pored značajnog broja novih operacija, poboljšan je i sam interpreter. Postoje posebne verzije za rad u Windows okruženju i pod DOS-om. Windows verzija može biti vrlo interesantna širem krugu korisnika, s obzirom na pomalo grub (ali upotrebljiv) korisnički interfejs DOS verzije, što je posebno neprijatno: mali broj drajvera za štampače i grafičke kartice. Međutim, treba imati u vidu da su Windows verzije 32-bitnih programa obično nešto sporije od DOS verzija.

Kakav je programski jezik *Matematike*? Često se može čuti (ili pročitati) kako je programiranje na *Matematici* slično ili isto sa programiranjem na C-u. To je čak delom i tačno utoliko što *Matematika* poseduje naredbe **Do, While, For, Break, Block, Return**, +=, *=, -=. S druge strane, *Matematika* nudi i čitav niz „neobičnih" operacija poput **Thread, Map, Apply, &, Rule, Inner**, pomoću kojih je moguće praktično uvek izbeći korišćenje klasičnih struktura. Ortodoksno gledište kaže da je izbor šta će od ovog programer koristiti potpuno nebitno, pošto je dozvoljeno da se koristi sve što sam jezik dozvoljava. U praksi tipična situacija je da su programi *Matematike* „mešavina" – kontrolne strukture (ljudi koriste po navici) i „neobične" operacije jer se bez njih često ne može. Pregled tako dobijenog koda obično izaziva glavobolju kod početnika (posebno ako veruje da je dovoljno poznavanje C-a za pisanje na *Matematici*) i mučninu kod istinskih ljubitelja *Matematike* (jer se dobijeni kod – ponekad jednostavnim transformacijama – može znatno kraće, elegantnije pa i korektnije zapisati). Kako onda pisati programe na *Matematici*? Na ovo se može odgovoriti tek ako se razjasni pitanje stila programiranja.

PITANJE STILA

Stil programiranja (neformalno gledano) predstavlja skup osnovnih koncepata zajedničkih za neku grupu programskih jezika. Program napisan *proceduralnim* stilom (FORTRAN, C, C++ ...) sastoji se od niza *naredbi* koje deluju na *promenljive* menjajući im vrednost ili utičući na redosled izvršavanja naredbi. Mada je ova ideja jednostavna i razumljiva, nije i jedina moguća. Tako u *funkcionalnom* stilu programiranja (najpoznatiji predstavnik je LISP) ne možemo promenljivama da promenimo jednom dobijenu vrednost, ne postoje čak ni naredbe koji bi se „izvršavale". Umesto toga, program predstavlja jedan izraz (funkciju) koja se interpretira prema jasno formulisanom skupu pravila. Pri tom nema eksplicitne razlike u načinu predstavljanja podataka i samog programa. To omogućava, recimo, da interpreter jezika bude (potpuno) opisan samim jezikom. U *logičkom* stilu programiranja „program" je skup iskaza, a „izvršavanje" pronalaženje (precizno definisanim postupkom) vrednosti *slobodnih* i *vezanih* promenljivih. U poslednje vreme, naročito sa pojavom C++ jezika, postaje vrlo popularno i objektno programiranje. Na žalost, objektno-orijentisani stil programiranja još nije dovoljno *razvijen*. Mada postoji nekoliko razrađenih teorijskih koncepata (poput teorije *komunicirajućih sekvencijalnih procesa*) koji bi „uz dosta prijateljskog razumevanja" mogli da se povezu sa objektno orijentisanim stilom, glavni problem još uvek predstavlja nedostatak čistih objektno-orijentisanih programskih jezika.

Stil programiranja, određujući osnovne postavke nekog programskog jezika, utiče i na takve osobine kao što su mašinski jezik i arhitektura računara na koma programi treba da se izvršavaju. Iz svega navedenog sledi da je pitanje stila programiranja zapravo osnovno pitanje kod svakog novog programskog jezika.

Očigledna je namera autora *Matematike* da se omogući potpuna primenljivost proceduralnog i delimična funkcionalnog stila. Međutim, osnovni koncepti ovog jezika ipak ne mogu da se uklope ni u jedan od gore pomenutih stilova programiranja. Stoga na *Matematiku* treba gledati kao na kompletno nov programski jezik sa nizom prednosti i ponekom manom u odnosu na druge jezike.

PRVI KORACI

Program u *Matematici* predstavlja neki izraz izgrađen od *simbola* i tzv. *sirovih podataka* (brojeva, na primer). Interpretacija (izvršavanje, izračunavanje) se sastoji iz menjanja zadatog izraza prema skupu transformacija sve dok se ne dobije izraz koji ne može dalje da se promeni. Programiranje se sada svodi na definisanje samih transformacija. Za razliku od drugih programskih jezika gde je varijablama neophodno (na neki način) dodeliti vrednost da bi je imali, simbol uvek ima vrednost – predstavlja baš samog sebe. Kao prvi primer uzmimo izračunavanje izraza

p=q; q=p;

C-ovski gledano čini se da će posle izvršavanja ovog izraza oba simbola imati vrednost **q**. Umesto toga, *Matematika* neće moći da izračuna izraz, ukazujući na pojavu beskonačne petlje. Ovaj naoko začuđujući rezultat je lako razumljiv ako se ima u vidu da se gornjim izrazom ne dodeljuju vrednosti simbolima, već definišu transformacije – **p** se preslikava u **q**, a **q** se preslikava u **p** i eto beskonačne petlje. Navedimo jedan realniji primer koji ilustruje ovu specifičnost – sledeći izraz bi trebalo da razmenjuje vrednosti simbola **z** i **w** koristeći se (pomoćnim) simbolom **b**:

b=z; z=w; w=b;

Čitaocu ostavljamo da pokaže (ili proveri) da u određenim situacijama navedeni izraz *ne* razmenjuje vrednosti simbola. Ovo je dovoljna ilustracija da se uoči potreba razlikovanja čak i sintaksno identičnih konstrukcija u jezicima različitog stila programiranja. Razmotrimo malo preciznije strukturu *Matematikinih* izraza.

SINTAKSA...

Izraz, kao osnovni i jedini objekat jezika, gradi se od simbola (**x1, Sin** ...) i konstanti (1, „a" ...) koje ćemo zvati atomima. Izraz je ili atom ili oblika **f[x,y, ..]** (gde su **f** (tzv. glava), **x, y, ..** (tzv. elementi) takođe neki izrazi). Time je definisana struktura proizvoljnog *Matematikinog* izraza.

U praktičnom radu, međutim, postoji niz sintaksnih olakšica pri zadavanju izraza. Npr. **x r y**, gde je r jedan od *Matematikinih* unapred definisanih simbola-binarnih operacija (popul. +, /, ==, ->, ...) takođe definiše izraz, uz mogućnost standardne upotrebe malih zagrada za eksplicitno određivanje načina vezivanja izraza binarnim operacijama. Zapis {**x,y,** ...} je skraćeni zapis izraza **List[x,y,** ...**]**. Zapisu **f[x]** ekvivalentni su **f@x** i **x//f**, a zapisu **f[x,y]** ekvivalentan je **x~f~y**.

Komandom **FullForm** se lako dobija stvarni (prefiksni) oblik svakog *Matematikinog* izraza. Tako će **FullForm[a=b/c]** dati **Set[a,Times[b,Power[c,-1]]]**.

U verziji 1.2 programa izraz je mogao da bude i oblika (**x,y,...**). Tada su **f@(x,y,...)** i (**x,y,...**)//**f** predstavljali moguće zapise izraza **f[x,y,...]**. Ova čudna jezička konstrukcija više izraza bez glave (tzv. *sekvenca*) je u verziji 2.0 dobila prirodniji oblik postavši **Sequence[x,y,** ...**]**, dok je način upotrebe ostao isti.

Ne treba da čudi što ne postoje nizovi, liste, strukture ... bilo koji tip se može predstaviti nekim izrazom.

... I SEMANTIKA

Neformalno govoreći, pod semantikom nekog (programskog) jezika podrazumevamo mogućnost da se za svaki program zapisan na tom jeziku dokaže da je ekvivalentan algoritmu koji predstavlja.

Čuveno struktuirano programiranje je bilo upravo pokušaj da se programima pisanim u proceduralnom stilu da semantika. To se, kao što je poznato, postiže upotrebom kontrolnih struktura poput **while** i zbegavanjem **goto** naredbe, a zatim primenom odgovarajućeg formalizma dokazivanja korektnosti. Programi funkcionalnog stila imaju najrazrađeniju semantiku – toliko razrađenu da se algoritam obično identifikuje sa odgovarajućim skupom LISP funkcija.

U praktičnom radu pitanje korektnosti programa ne mora biti naročito bitno – važno je da jezik omogućava što lakše pisanje i testiranje programa. Ako pri tom jezik ima semantiku dokazivanja korektnosti za program koji „očigledno" radi može biti „istenvanje strogosti". *Matematika* (koliko mi je poznato) još nema razvijenu semantiku u gore opisanom smislu. Napisani programi jednostavno lepo rade i to je sve.

Osnova izračunavanja u *Matematici* je transformisanje jednog izraza u drugi prema zadatim (ili ugrađenim) transformacionim pravilima. Nije preterano zamisliti *Matematikin* interpreter kao neku vrstu makro-procesora izraza.

Transformacije definišemo pomoću **Rule** i **RuleDelayed** operacija (u infiksnom obliku predstavljenih simbolima -> i :>) izrazom sledećeg oblika

a->b ili a:>b

Ovim se označava transformacija kojom se izraz opisan izrazom **a** zamenjuju izrazom **b**. Razlika između ove dve operacije je u sledećem: dok se kod upotrebe **Rule** vrednosti izraza **b** izračunava se u trenutku navođenja transformacije, upotreba **RuleDelayed** izračunavanje desne strane odlaže do trenutka njene neposredne primene.

Izraze opisujemo ili navođenjem konkretnog izraza kog transformacija menja ili opisivanjem čitave klase izraza pomoću *izraza uzorka* (*pattern*-a). Ključni simbol je **Blank** koji opisuje (poput džoker znaka) proizvoljan izraz. **Blank[a]** (skraćeno **a_**) predstavlja proizvoljan izraz koji se nadalje (lokalno u definiciji transformacije) pridružuje simbolu **a**. Simboli **BlankSequence** i **BlankNullSequence** (skraćeno __ i ___) opisuju redom sekvencu od jednog ili više, odnosno sekvencu od nula ili više izraza. Na primer, izraz **f[1,a]+g[0]** pripada klasi izraza opisanih uzorkom **f[a__]+x_**. Sam postupak prepoznavanja da li izraz zadovoljava neki uzorak se naziva *pattern matching* (najbliži analogon iz drugih programskih jezika je *unifikacija* iz PROLOG-a). Moguće je eksplicitno za

[illegible] ...opisivanju izraza da izraz koji predstavlja neku [illegible] zadati oblik pomoću **Pattern** i **PatternTest** [illegible] u pomoćnom zapisu odgovaraju im simboli i [illegible] i __?VectorQ} u izrazu uzorku zahteva se [illegible] bude lista od jednog ili više vektora.

Transformacije primenjujemo pomoću **ReplaceAll** i **ReplaceRepeated** operacija (/ odnosno // u infiksnom zapisu), izrazom oblika e / l odnosno e // l gde je l transformacija koju treba izvršiti. Prvom konstrukcijom se primenjuje transformacija na izraz, dok se drugom primena transformacija vrši uzastopno sve dok izraz ne prestane da se promeni. Moguće je zadati listu (ili čak [illegible]) transformacija – tada se transformacije primenjuju [illegible] kojim su navedene u listi. Na primer, sva [illegible] distributivnog zakona množenja prema sabiranju u izrazu e možemo da izvršimo izrazom

a_(b_+c_)-> a b + a c

Uslovne transformacije tj. transformacije koje se [illegible] obzir samo ako je vrednost nekog izraza kon[illegible] [illegible] definiše se **Condition** operacijom (infiksno [illegible]) oblika

[illegible] tj odnosno e // t /; q

[illegible] q zadati uslov. Tako se izračunavanje funkcije [illegible] u izrazu e može izvesti pomoću

sgn(x_)->0, sgn(x_)->1/;x>0, sgn(x_)->-1/[illegible]

[illegible] navedenim konstrukcijama (uz neka neophodna [illegible] postupka interpretacije (što će biti dato kasnije) moguće je realizovati bilo kakav algoritam. [illegible] Transformacija može da se učini globalnom pridruži[illegible] nekom simbolu pomoću nekoliko operacija poput **Set** i **SetDelayed** (= i :=) uz moguću upotrebu **Condition** operacije na praktično isti način kao kod **Rule** operacije. Pridruživanjem se transformacija izvodi svaki put [illegible] uzorak koji se transformiše pojavi u interpretiranom izrazu. Uz simbol možemo vezati više transformacija, [illegible] redosled kojim se primenjuju odgovara (uz nekoliko [illegible]) redosledu kojim su definisane.

[illegible] simbolu se daje može dodeliti niz atributa što utiče [illegible] primene transformacija vezanih za dati simbol. [illegible] ilustraciju pretpostavimo da se transformacija izraza [illegible] ne menja pri zameni mesta argumentima. Ovu [illegible] ne možemo opisati prostim navođenjem [illegible] f[y,x] jer bi to dovelo do beskonačne petlje. [illegible] toga, pokrebno je simbolu f dodeliti atribut **Orderless** sa

Attributes[f]={Orderless}

[illegible] svaki put kada se pojavi izraz **f[x y]** pre [illegible] odgovarajuće transformacije argumenti će biti [illegible] rastući redosled. Slično je moguće tretirati i [illegible] asocijativnost transformacije – cela stvar funkcioniše na principu svođenja izraza na neku vrstu kanoničkog oblika za datu osobinu. Na žalost, nije moguće [illegible] sopstvene atribute (npr. za antisimetrične transformacije) što je nedostatak koji bi trebalo otkloniti u [illegible] verzijama programa.

[illegible] Interpreter pre izvršavenja neke transformacije obično izračuna vrednosti argumenata te transformacije, [illegible] ponekad je neophodno da argument bude [illegible] transformaciji neizračunat. Ovakve transformacije se nazivaju *specijalne forme*. Tako je **RuleDelayed** specijalna forma, dok **Rule** nije. Tri atributa određuju [illegible] forme: **HoldFirst**, **HoldRest** i **HoldAll**: a transformacija kojoj je pridružen neki od ova tri atributa [illegible] izračunati redom prvi, sve sem prvog, odnosno sve argumente. Specijalne forme (prvi put realizovane [illegible]) predstavljaju analogon kontrolnih struktura iz proceduralnih jezika (interesanto je da se npr. i u C-u [illegible] definisati nove kontrolne strukture pisanjem odgovarajućeg preprocesora koji bi prevodio datoteke sa proširenjima pre kompajliranja u običan C program. FORTH [illegible] mogućnost definisanja kontrolnih struktura).

[illegible] je moguće približno skicirati postupak izračunavanja vrednosti nekog izraza. Ako je izraz simbol bez [illegible] transformacije ili konstante, rezultat je isti taj [illegible] Suprotnom, prvo se izračunava vrednost glave e [illegible] vrednost elemenata izraza (koristeći se rekurzivno [illegible] postupkom) uz mogućnost da, ako su **Hold** [illegible] prisutni, neki od elemenata izraza ne bude [illegible] Daje se u zavisnosti od ostalih atributa glave [illegible] svođenje izraza na konačni oblik za svaki atribut glave [illegible] jednačno. Tek sada se proverava da li postoji neka transformacija korisnika čiji uzorak zadovolja[illegible] izraz (a zatim se ista provera vrši i za *built-in* [illegible]), ako postoji izvršava se istim postupkom [illegible] na izraz dobijen primenom transfor[illegible]

Kako se u opisani postupak interpretacije uklapaju konstrukcije poput **While** ili **For**? Odgovor je – nikako. Potrebno je posebno definisati pravila interpretacije za svaku kontrolnu strukturu ponaosob. Takođe i neke sasvim „obične" operacije poput x++ moraju biti posebno definisane (pošto će x=x+1 proizvesti beskonačnu petlju) što govori da su proceduralne konstrukcije zapravo veštački ugrađene u jezik.

Funkcionalni stil programiranja je već mnogo bliži prethodnom postupku izračunavanja, ali s druge strane nepostojanje tzv. odloženog izračunavanja (*lazy evaluation*) i nekih drugih osobina čini da je strožije gledano funkcionalni stil samo delimično upotrebljiv. Postoje doduše tzv. čiste funkcije (*pure function*) kao analogoni lambda funkcija iz LISP-a, ali one ne zadovoljavaju stroge uslove funkcionalnog programiranja. Njih treba shvatiti prvenstveno kao još jedan od načina da se zadaju transformacije. Treba pomenuti i elegantno rešen način opisivanja ovih funkcija (pomoću simbola # i postfiksnog operatora &), veoma praktičan u mnogim realnim situacijama (mada neprimenljiv u opisivanju funkcija višeg reda).

Listing 1

```
Iter [x_]   :=Hold[ [illegible] ]   iteraija
                [illegible]          iteraija > e

Attributes[Iter] [illegible]

trules = R [illegible] De ay [illegible]

Local [body_ defs_ t u es [illegible] utoam HoldBody [illegible]

Attributes[Local] = { o d [illegible] p e

(* Primer *)

f[x_] =Local[q q   q >x x]   q=1,  Print[f[a]]
```

Specijalne forme Local [expr, rule1, rule2, ...]

GLOBALNO I LOKALNO

Opisani način na koji *Matematika* definiše transformacije i izračunava izraze jednostavan je i moćan, ali nije do kraja razrađen. Neki problemi nastaju usled pomalo nejasno definisanih domena važenja imena pojedinih simbola. Da se i sam interpreter (u verziji 1.2) ponekad ume da zbuni pokazuje slika 1. Zatraženo je izračunavanje istog izraza dva puta, samo sa jednim različito imenovanim simbolom. U oba slučaja dobijen je na kraju tačan rezultat, ali je u jednom slučaju dobijeno i nekoliko poruka (sa aspekta korisnika potpuno irelevantnih). U verziji 2.0 *Matematike* ovaj nedostatak je otklonjen.

Problem koji i dalje ostaje je nemogućnost definisanja *lokalnih* transformacija. Ovo ćemo ilustrovati na sledećem sasvim jednostavnom primeru transformacije koja računa četvrti stepen nekog izraza

f[x_] = q q / q->x x

Pisanje i testiranje ovakve funkcije zahteva dvadesetak sekundi rada. Uprkos tome, može se desiti da funkcija u nekom kasnijem trenutku prestane da daje željeni rezultat. I zaista, ako definišemo globalnu transformaciju **q=2**, funkcija će početi da za svaki zadati argument vraća četvorku kao rezultat. Da bi uočili uzrok nepravilnog rada, razmotrimo postupak izračunavanja izraza **f[2]**. Interpreter prvo, upoređujući zadati izraz sa zapamćenim transformacijama, uočava da izraz zadovoljava uzorak sa leve strane definicije od f, x dobija vrednost 2, a zatim se izračunava **ReplaceAll** tako što se izračuna vrednost levog i desnog izraza a onda primeni transformacija. Međutim, vrednost levog izraza je, pošto je q=2, jednaka 4 i samim tim transformacija neće biti izvršena. Logična pretpostavka je da bi, ako bi transformacija bila primenjena na neizračunat izraz **q*q**, cela stvar radila korektno. Dve specijalne forme su nam pri tom potrebne: operacija **Hold[e]** sprečava izračunavanje izraza **e**, dok operacija **Literal[p]** sprečava izračunavanje uzorka **p** omogućavajući *pattern matching* pre nego što se izračuna uzorak. Dakle, sve što treba učiniti je izračunati vrednost izraza dobijenog primenom transformacije na neizračunat izraz **q*q**. Operacija **Evaluate** vrši računanje vrednosti zadatog izraza (iza nje se krije ceo interpreter!). Stoga bi funkcija mogla da izgleda

f[x_] = Evaluate[(Hold[q*q]/.Literal[q]->x x)]

Prirodno je pitanje da li se opisanim postupkom može proizvoljna (lista) transformacija učiniti lokalnom? Odgovor je, na žalost, ne. Ovako se mogu učiniti lokalnim samo transformacije koje pojedinačne simbole preslikavaju u izraze ili definicije nerekurzivnih transformacija koje se pri tom pozivaju sa argumentom čije izračunavanje nije potrebno. Čak i tako smanjen skup lokalnih transformacija može biti vrlo praktičan (njime se izbegava upotreba **Block** operacije za definisanje lokalnih varijabli). Stoga je definisana (listing 1) specijalna forma **Local[e,t1,...]** koja izračunava vrednost izraza **e** koristeći se datim lokalnim transformacijama **t1**, **t2**, ... Uz upotrebu ove specijalne forme lako je dati novu definiciju transformacije **f**

f[x_] =Local[q q, q->x x]

Moguće je napisati i specijalne forme (nazovimo ih po analogiji **LocalRule** i **LocalRuleDelayed**) koje bi dozvoljavale definisanje proizvoljnih lokalnih transformacija (što je korektnije rešenje od korišćenja tzv. *konteksta*, mada manje operativno). Pisanje ovih formi nije pravolinijsko kao u prethodnom slučaju jer je potrebno imitirati kompletnu interpretaciju izraza. To, s druge strane, nije preterano težak posao (ne treba zaboraviti da je interpreter uvek na raspolaganju preko **Evaluate** operacije), ali, posebno u praktičnom radu, pomenute forme ipak treba da budu *built-in*.

Upravo iz tih razloga trenutno je u ispitivanju interpreter za *Matematiku* napisan na (funkcionalnom) LISP-u. U nekoliko stotina linija obuhvaćena je sintaksna analiza, *pattern matching* i interpreter *hard core* operacija *Matematike*. Razume se, prvi entuzijasta bi ceo posao mogao da uradi i na C-u (ili C++), ali uz više linija koda, više utrošenog vremena na razvoj, više tehničkih i teorijskih problema.

Navedimo na kraju i jednu zanimljivu osobenost *Matematikinih* izraza. Neka je definisano f=2, x=a, a zatim izračunat izraz **f[x]**. *Matematika* daje kao rezultat **2[a]** (nezavisno od postojanja gornje definicije za **f[x_]**). Dobijeni izraz, doduše, ne protivreči izloženim principima rada *Matematike*, ali nije u skladu sa (najčešćom) interpretacijom izraza **f[x]** kao pozivom funkcije f sa argumentom x. Treba primetiti da ovakve situacije *ne bi* mogle da se eliminišu prostom zabranom da vrednost glave bude neka konstanta, jer tada (s obzirom na slobodu da se redom i sa još nedefinisanim transformacijama ili čak da se dinamički promeni značenje već definisanih transformacija) nikad ne bismo bili sigurni da je bilo koji izraz sa glavom ispravan. Drugi mogući pristup je da se uvedu *dve* vrste simbola, što je krajnje neprirodno rešenje.

POČETAK JEDNOG DIVNOG PRIJATELJSTVA

Elegantna sintaksa, moćne operacije i bogata biblioteka raznih funkcija omogućavaju da i vrlo komplikovana izračunavanja ili definicije budu zapisani u svega nekoliko redova ekrana. U samom programskom jeziku *Matematike* najviše oduševljava ideja potpune simboličkog tretiranje svakog izraza. Svaki simbol, ma koliko bio okićen transformacijama, atributima ili opcijama, ostaje simbol. Kao primer može da posluži transformacija (iz listinga 1) **trules** ___**Rule** | ___RuleDelayed. Teško je naći analogon ili bar opisati ovakvu konstrukciju terminima nekog drugačije koncipiranog programskog jezika. Možda „Varijabli je dodeljen tip izgrađen od rezervisanih reči..." Prilično besmisleno, zar ne?

Koncentrišući se na osobenosti *Matematike* kao programskog jezika, nije bilo moguće detaljnije razmotriti stotine raznovrsnih *built-in* i eksternih operacija koje joj dodatno povećavaju moć, a mogu i same po sebi da budu odlučujući razlog za korišćenje ovog paketa. Nije bilo moguće ni na realnim primerima ilustrovati osobenosti programiranja na *Matematici*. Takođe bilo bi interesantno uporediti osobine programa na *Matematici* pisanih različitim stilovima, testiranje ugrađenih operacija, merenje performansi programa... U svakom slučaju, ostaje nam izreka da je jedini način da se nauči neki programski jezik – pisanje programa na njemu. Primenjeno na *Matematiku* to može predstavljati pravo zadovoljstvo.

VI STE SASVIM O.K.



SAMO NEKE STVARI NISU PO VAŠOJ MERI

Zato Vam nudimo grafiku visoke rezolucije:

KOLOR MONITOR MAG MX17F

MAG je sedamnaestoinčni multi-skening kolor monitor ultra-visoke rezolucije (1280 x 1024 piksela N I) sa interaktivnim LCD displejom. Kompatibilan je sa postojećim grafičkim standardima i raspoznaje grafičke modove. Poseduje BNC konektor za prihvat signala sa profesionalnih procesorskih grafičkih adaptera. Veličina rastera tačke je 0 26 mm. Dizajn omogućava preciznu kontrolu, autosihronizaciju i automatsko podešavanje veličine ekrana. Korisnički je orijentisan, programibilan i fleksibilan.



OTC - COMPUTERS je prva tajvansko - jugoslovenska kompanija koja se bavi proizvodnjom i prodajom PC računara. Pomažemo Vam da profesionalno ostvarujete vaše zamisli i da stalno povećavate kriterijume svoga poslovanja. Takva su vremena. Pozovite nas. Rado ćemo vam pružiti dodatne informacije.

OTC
COMPUTERS

MILENTIJA POPOVIĆA 9 • S. CENTAR • N. BEOGRAD • TEL. 381 11 222 43 22 EXT 684 • 381 11 222 23 55

LAKO MODULISANI ROCK 'N' ROLL

Već nekoliko godina u svetu postoji trend koji se zove – moduli. To su fajlovi u kojima se kriju notni tekst, dirigent i čitav orkestar. Otkako su zvučne kartice i kod nas u široj upotrebi, moduli imaju sve više poklonika.

Josip Medli

Moduli su fajlovi koji u sebi sadrže samplove informacija o semplovima, parametre za emitovanje i raspored kojim se emituju. Najčešće nose ekstenziju MOD. Kada se sve to prevede na ljudima razumljiv jezik, dobija se ceo orkestar sa notnim tekstovima i dirigentom u jednom fajlu. Zahvaljujući toj kompaktnosti (sva potrebna informacije su na jednom mestu), moduli su brzo postali standard za kompjuterski generisanu muziku na svim platformama – od PC-ja do radnih stanica.

Moduli su stekli veliku popularnost. Komponovana pesma može vrlo lako putovati i to ne radiotalasima, već modemskim vezama. Ne na pločama i kompakt diskovima nego u komprimovanom obliku, od kompjutera do kompjutera, nekoliko minuta potpuno besplatno.

POČELO JE DAVNO

Glavni "krivci" za ovu muzičku revoluciju su dva momka koji se kriju iza pseudonima Mahoney i Cactus. Oni su davne 1987. napravili program za komponovanje i nazvali ga *SoundTracker*. Pored ostalih prednosti koje je doneo ovaj program, najvažnija je bila mogućnost snimanja pesme kao modula. To je ova dva momka proslavilo i zlatnim slovima ih upisalo u istoriju kompjuterske muzike.

Nakon *SoundTracker*-a pojavili su se mnogi programi iste namene i svi su u svom imenu nosili ono "Tracker" (*NoiseTracker, GalaxyTracker, ProTracker*). Zato se danas muzički programi koji koriste MOD format nazivaju i "trekeri".

Masovno uživanje u MOD pesmama na PC računarima omogućeno je pojavom SoundBlaster-a i sličnih zvučnih kartica. Ali tek je SoundBlaster PRO sa svojim stereo zvukom bio dostojan konkurent originalu, a o 16-bitnim karticama da i ne govorimo. (Ukoliko vas zanimaju detaljnije tehnička karakteristike zvučnih kartica, otvorite "Računare 94" na strani 54.) Kako su godine prolazile, pojavljivali su se novi programi za pravljenje modula ("trekeri") i svaki od njih je ubacivao nešto novo u MOD format, tako da danas imamo veliki broj standarda.

SoundTracker i *NoiseTracker* su pioniri MOD pesama. Koriste samo 4 kanala, ali su u vreme u kojem su nastali bili savršenstvo. Uporedo sa ovim formatima na PC-ju je nastao i STM format, ali je njegova slava kratko trajala.

ProTracker nastavlja tradiciju 4 kanala, ali kvalitet ispravlja velikim brojem efekata i komandi za puštanje semplova. *ProTracker* je danas najzastupljeniji MOD format, a po svemu sudeći tako će i ostati neko vreme.

Octamed i *669* unose značajnu promenu – 8 kanala. Poboljšanjem kvaliteta, standard se dovodi u pitanje. *Octamed* je nastao na Amigi, a 669 na PC-ju i, iako se mogu slušati na oba kompjutera, ljudi se retko odlučuju za "strani" format. Pošto su se pojavili nedavno, nemamo dovoljno informacija o njihovom kvalitetu, pa ćemo ih ovoga puta zaobići.

Naravno, ovo nisu jedini formati koji postoje. Još desetine i desetine su razvijane na osnovu njih, ali ili nisu bili dovoljno dobri te su zaboravljeni, ili se veoma malo razlikuju od originala, pa nije bilo svrhe pamtiti ih.

Da uprostimo, semplovi su "instrumenti" zahvaljujući kojima možemo da čujemo MOD pesme. U modulu je svaki sempl obeležen brojem između 0 i 31. Da bi kompjuter znao kojim redom i kojom brzinom da pušta te semplove, potrebno je još nekoliko informacija. Te informacije su raspoređene kroz *pattern*-e. *Pattern* znači obrazac, ali ga je u ovom slučaju bolje shvatiti kao stranu notne sveske. U jednom modulu moguće je imati do 100 takvih strana koje su obeležene brojevima od 0 do 99. Svaki *pattern* podeljen je na četiri kanala (dva leva i dva desna), a svaki kanal na još 64 dela (ili 64 note). Nota u sebi sadrži četiri informacije – tonalitet, broj sempla (instrumenta), efekat i vrednost efekta.

Osim informacija koje se nalaze u *pattern*-ima, postoje i informacije o redosledu njihovog puštanja. Znači, nismo ograničeni na puštanje redom, od prvog do poslednjeg, nego su moguće i kombinacije tipa prvi, drugi, drugi, osmi, peti, prvi.

Semplovi koji se koriste u modulima su najčešće mono, 8-bitni i to semplovani učestalošću od 15 do 22kHz – ni na pola puta do CD kvaliteta. To naravno nije dovoljno za profesionalnu produkciju, ali u kućnim uslovima nimalo ne smeta. Razlog za loš kvalitet modula je skromna oprema na kojoj se prave – SoundBlaster, SoundBlaster PRO ili *low-end* modeli Amige, najčešće.

Ipak, kako tehnologija napreduje, pojavljuju se i tehnički savršeni semplovi – CD kvaliteta, ali ipak retko, jer moduli sa takvim semplovima zauzimaju mnogo više mesta. U prvom slučaju (lošiji kvalitet) semplovi su prosečne dužine između 200 i 300 K, dok su u drugom (CD kvalitet) dugački i nekoliko megabajta.

Nadamo se da vam je ovo pomoglo da u glavnim crtama shvatite o čemu se radi. Ukoliko nije, u nastavku ćete naći informacije koje će vam koristiti u svakodnevnom slušanju modula.

'OĆU DA IH SLUŠAM!

Da biste slušali module, potreban vam je program koji će ih puštati: *Module Player*. Takvih programa ima mnogo, ali samo par njih su dovoljno dobri i pouzdani da bi se mogli koristiti. Mi vam preporučujemo program po imenu *Dual Module Player* (DMP) i to od verzije 2.70 pa nadalje.

Dual Module Player podržava sve standardne zvučne kartice: Sound Blaster, SoundBlaster PRO, SoundBlaster 16, Pro Audio Spectrum+, Pro Audio Spectrum 16, ARIA, Windows Sound System kompatibilne, Gravis Ultrasound, razne "home made" kartice i COVOX, a ukoliko baš morate, module možete slušati i preko originalnog PC speaker-a. Pored toga za svaku od tih kartica postoji puno dodatnih parametara koji mogu poboljšati ili pogoršati zvuk, u zavisnosti od potrebe – na stereo karticama možete isključiti stereo, na 8-bitnim pojačati visoke tonove, na mono karticama uključiti *surround system* pa dobiti veštački stereo, i tako dalje.

Osim impozantnog broja kartica na kojima se moduli mogu slušati, ovaj program podržava i veliki broj formata modula – AMF, STM, 669, S3M, MOD i NST. Od ovih formata aktuelni su MOD i 669, ali su ostali ostavljeni zbog nostalgije i kompatibilnosti.

Ukoliko imate modem, *Dual Module Player* – i to verziju 2.76 – možete naći na SEZAM-u u direktorijumu \IBMPC\FUN pod imenom DMP276.ZIP.

Mnogi korisnici se odlučuju i za *Visual Player* koji pre svega plení svojim izgledom. Radi u grafičkom modu u 256 boja. Ima razne merače zvuka (osciloskopa za svaki kanal, jačinu svakog kanala...) pa čak i *light show*. Svim opcijama se pristupa mišem. Jedini je nedostatak što traži malo jače kompjutere i više memorije da bi sve dovoljno brzo radilo.

Vremenom i upornim slušanjem ćete stvoriti predstavu i o ljudima koji komponuju module. Veći na ih nije muzički obrazovana – takvi "kompozitori" se dokazuju preveći dosadne tehno numere u kojima dominira bas bubanj. Oni malo iskusniji i dalje imaju afinitete prema tehno muzici, ali na višem nivou (neke njihova numera bi sigurno uspele i na MTV-ju). Kompozitori koji drže do sebe trude se da zaobiđu tehno, pa probaju sa raznim akustičnim gitarama, ambijentalnom muzikom, velikim brojem efekata, gitarskim solo deonicama. Takođe važna stvar je i dužina modula – ako je modul kratak i efektan, definitivno skuplja simpatije "poznavaoca".

Još jedan savet: ukoliko ste u mogućnosti, povežite zvučnu karticu sa nekim jačim pojačalom ili stubom. Razlika u odnosu na male i bedne zvučnike koji se obično dobijaju uz karticu je neopisiva.

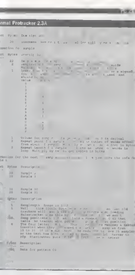
'OĆU DA IH PRAVIM!

Kao što je potreban softver za slušanje modula, potreban je i softver za njihovo komponovanje. Verovatno najrasprostranjeniji program te namene za PC računare je *MODEdit* (i on se može naći na SEZAM-u). Rad sa njim je savršeno lak, a uz to ima i odlično uputstvo. Sve se svodi na rad sa mišem i kliktanje na opcije u menijima, a tastatura je potrebna samo kada treba otkucati/odsvirati neku notu.

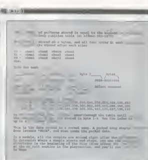
Proces komponovanja modula je identičan sa radom pravih muzičara u studiju. Krene se sa ritmom, s tim što ovde bubnjeve nećemo tegliti, nego ćemo ih učitati u memoriju – deo po deo. Uz *MODEdit* se dobija nekoliko semplova koji su dobri za početak. Prvo bas bubanj (*Bassdrum, Kick*), onda doboš (*Snare*), pa činele (*Hihat, Crash, Ride*) i možda koji prelaz (*Tom*). Kada ste učitali sve semplove koji su potrebni za ritam, treba odsvirati tj. odlupati željenu deonicu. To je vaš posao.

Ukoliko već znate kako pesma treba da zvuči, odmah dodajte i bas gitaru. Ali jednostavnije je da prvo smislite temu (ono po čemu će ljudi pamtiti vašu pesmu), odsvirate je na instrumentu koji izaberete i onda, slušajući tu temu, komponujete bas deonicu. Za ovakav primer dovoljan je jedan *pattern*. Ako ste ambiciozniji, kakvi bi trebali da budete, broj *pattern*-a će se povećavati.

Ukoliko odlučite da nastavite sa komponovanjem, trebaće vam i vaša kolekcija semplova. Do-



Format Protracker 2.3A



Ukoliko želite da napravite program koji će moći da učitava ili snima MOD pesme, evo formata u kojem se zapisuju informacije (*Protracker 2.3A*, što će reći standard)

Top lista modula

Mesto	Modul	Dužina	Vreme	Ime fajla
1	„Elektfunk!"	219 k	3:15	elektnk.mod
2	„Das Boot"	351 k		dasboot.mod
3	„Sea Of Love"	177 k	3:41	seaolove.mod
4	„What is NORML?"	134 k	4:35	norml.mod
5	„Maximum Overdrive"	453 k	4:00	max.mod
6	„Galep Obretkovich"	238 k	2:30	galep.mod
7	„1 2 3 4 5 6 7 8 9"	129 k	4:22	12345678.mod
8	„Guitar Slinger"	406 k	4:45	gslinger.mod
9	„Badava ti higijena"	305 k	2:34	higijena.mod
10	„Classic Techno Mix 2"	314 k	4:30	clasic2.mod

Top listu modula slušajte svake subote od 17 do 18 časova u emisiji "Modulacije" Radio-Pingvina, na 90.9 MHz

datne semplove možete sami snimiti (skoro svaka zvučna kartica ima ugrađen sempler) ili ih "izvaditi" iz drugih modula. Vodite računa da ne preterujete sa "vađenjem" jer takvi stvaraoci nisu cenjeni.

MODEdit podržava manji broj kartica nego *Dual Module Player*, ali su glavne ipak tu – SoundBlaster, Disney Sound Source, Covox i opet PC speaker. Moduli se snima u najstarijem *Sound Tracker* formatu. Registrovanjem programa dobijate nova opcija, kao što su direktno snimanje melodije sa tastature (koja u ovom slučaju imitira klavijaturu), automatizovanje nekih poslova, podrška za EGA/VGA kartice i ostale sitnice koje život znače.

TOP-LISTE

Module najčešće prave kompjuterski entuzijasti koje se kriju iza različitih pseudonima. Svoje radove šalju na BBS ove i mreže i tako se vremenom proslave. Pošto je konkurencija uvek velika, za nekoliko godina postojanja, moduli su došli do kvaliteta koji je ravan, ili čak i bolji od zvaničnih izdanja na pločama i kompakt diskovima. Neki od najcenjenih autora su Andamar, Sidewinder, Audiomonster...

Takav trud zaslužuje priznanje, pa se često organizuju i muzička takmičenja gde su nagrade prilično visoke i korisne (par hiljada dolara, kompjuteri, dodatni hardver). Sve to je dovelo do stvaranja zvanične svetske top liste modula (MODCHART) koju je pokrenuo Mr. Oliver sa univerziteta u Muensteru, Danska. Glasovi za listu Mr. Oliveru stižu putem elektronske pošte. Svakih mesec dana glasovi se broje, sređuju i pravi se lista. Ona se distribuira preko Internet-a ljudima koji su se, uslovno rečeno, jer je cela usluga besplatna, pretplatili na nju i po konferencijama koje se bave muzikom. Glasovi za MODCHART stižu iz celog sveta – po informacijama koje smo dobili od Mr. Olivera, ima preko hiljadu glasova mesečno.

Od decembra 1993. godine i kod nas postoji top-lista modula i to u organizaciji Radio Pingvina i emisije "Modulacije". Ova lista se pravi u saradnji sa svetskom – nedeljno izdanje svetske mesečne top liste. Listu možete naći i na SEZAM-u, u konferenciji ORKA/Razno. Ukoliko već pravite module ili želite tek početi, svoje radove možete poslati kao predlog za ovu listu, i to elektronskom poštom preko SEZAM-a (za korisnika *jmodi*). Princip glasanja je sličan kao i na svetskoj listi, samo što je u domaćoj verziji moguće i glasanje putem klasične pošte. Ukoliko vas interesuje top-liste, svake subote između 17 i 18 časova uključite radio i namestite frekvenciju na 90.9 MHz FM. Imaćete šta da čujete!

GDE IH IMA

Moduli se najlakše distribuiraju modemskim vezama, pa se tako mogu naći na gotovo svakom BBS-u. Tu moduli borave u spakovanom obliku, najčešće PKZip i LHArc arhivema. Tako im se dužina značajno smanjuje, a samim tim je i prenos mnogo kraći. Sada ste spremni na sva iznenađenja koja vas mogu zadesiti.

Modula ima i na Internet-u. Ukoliko možete, probajte da svratite do sistema ftp.wustl.edu i pogledate u direktorijum /pub/aminet/mods. Tu ćete naći najbogatiju kolekciju modula na svetu, u svim formatima. Ako ne možete na Internet, tu je SEZAM. Prošetajte se kroz konferencije i uz neku od poruka ćete sigurno naći nešto zanimljivo. Moduli se često pojavljuju u konferenciji AMIGA/zvuk.

I još nešto za kraj. Često se postavlja pitanje da li je moguće konvertovati WAV, VOC ili SAM fajlove u MOD format. Odgovor je NE!

Prvo, razlika je u samim namenama – WAV, VOC i SAM su semplovi, dok je modul pesma koja u sebi sadrži nekoliko semplova. Naravno, nije tehnički neizvodljivo, ali nema potrebe to raditi. Kada bismo konvertovali modul dužine 200 K koji se sastoji od, recimo, 36 *pattern*-a, dobili bismo sempl vremenske dužine između tri i četiri minuta, a na hard disku bi zauzimao više megabajta.

BEZ PRAVOG FAVORITA

Zahvaljujući usavršavanju hardvera, tržište softvera je zapljusnuto novim šahovskim programima, ali njihovi algoritmi manje-više svi nalikuju jedan na drugog. Ovom prilikom dajem prikaz nekoliko manje poznatih programa.

Zoran D. Brkić

U prvim danima nastanka računarske groznice kod nas, ubrzo posle pojave ZX 80 i ZX Spectruma, pre nekih desetak godina, pojavio se i puno raznih programa, a među njima i priličan broj šahovskih. Nije ni čudo – misaone igre, a naročito šah kao najsloženija od njih, idealan su model za simuliranje rada čovekovog mozga. Tu se oduvek tražio odgovor na strateško pitanje – na koji način čovek donosi odluke. Eh, kad bi se za to moglo naći rešenje, bilo bi zatim jednostavno malo doterati mašinu, uložiti još samo malo kapitala, omogućiti joj da se razmahne njena kreativnost.

Kod PC računara, dugo se čekalo na prve šahovske programe. Prvo su se pojavili *Sargon* i *Zarkov* rađeni u nekakvom tekst modu, gde su konture figura bile izvedene uz pomoć „paragrafa". Zatim duga pauza, pa se pojaviše *Psion Chess* i prve verzije *Chessmastera* negde 1984. ili 1985. *Psion Chess*, iako se pojavio ranije, nadživeo je nekoliko verzija *Chessmastera*. Pokazao se kao jači igrač, iako je daleko zaostajao „po lepoti", kako po grafičkom izgledu tako i po opštoj „izvedbi". Zatim je nastupio duži period tapkanja u mestu, smenjivanja na tronu novih i novih verzija *Chessmastera* i tako kroz celu istoriju, sve do prošle godine. Teško je objasniti zašto je PC tako zanemaran. Možda zato što je njegov procesor smatran slabim u odnosu na Motorolin, jer nisu nedostajali šahovski programi za druge tipove računara? Možda zato što na Zapadu nije bilo dovoljno naklonosti za šah u „širokim narodnim masama"? Ko zna? Može se samo nagađati.

BUM NA PC PLATFORMI

Uglavnom, u poslednjih par godina taj se trend potpuno preokrenuo. Šahovski programi sustižu jedan drugog, nailaze nove verzije, kao da je započelo nadmetanje u tipično olimpijskom duhu – ko će bolje, brže, moćnije. Neki impuls je podstakao istraživanja u tom smeru, a kakav tačno, teško je odrediti. Postoje razne indicije. Procesori su postali dovoljno brzi da mogu proračunati milione operacija, te igraju jako „cug", a RAM memorije dovoljno velike i jeftine da drže sve varijante „u glavi", baš kao čovek tokom igre. Šah je dobio na popularnosti u Zapadnoj Evropi poslednjih godina. Nekad se porast popularnosti šaha na Zapadu objašnjavao stereotipno Fišerovom pobedom u meču nad Spaskim. Od tada je prošlo mnogo vremena, te su uzroci verovatno drugi. Možda je porast standarda stvorio višak slobodnog vremena, pa su se mnogi rešili da stvari doteraju do apsurda i traće vreme na takvu „zgubidansku igru" kao što je šah. Šah je naročiti „boom" doživeo u Nemačkoj, zemlji koja je toj igri dala mnoge velikane, ali ih se i odrekla zbog rasne čistote. Uložen je veliki kapital, kupljeno je mnogo jakih igrača, stvorena najjača liga u Evropi, a valjda će i amateri nešto naučiti. Organizuje se puno takmičenja, izdaje puno časopisa, pa čak postoje i usko specijalizovani koji se odnose isključivo na kompjuterski šah. Samo još fale vrhunski rezultati!

Ljudi kao ljudi, prihvatili su modu. I sada svaka država ima poneki šahovski program! Za svakog programera, pravi je izazov napraviti vlastiti program te vrste. Osim toga, takav je model vrlo poučan. Tu se vežbe i grafike (partiture u Asembleru!), zatim treba napraviti nepogrešivu logiku, pa meniji, pa rad s datotekama...

U ludoj trci koja je nastala, proizvođači su započeli da skidaju programe koji su se dobro pokazali kod šahovskih kompjutera i prebacuju ih na PC – računare. Za razliku od drugih vrsta programa, kod šaha je moguće snagu izmeriti sa priličnom tačnošću – pomoću rejtinga, kao kod živih ljudi. Problem je samo što se to ne radi na isti način. Proizvođač jednostavno proglasi da njegov program ima taj i taj rejting, deklarisani podatak prenesu u par časopisa, ako treba održi se i neko takmičenje kompjutera gde se okiti nekom medaljom pobednika. Sve podseća na one čuvene sajmove pića u Španiji, gde uvek pobeđuju neka naša vina! A pravilno bi bilo da se kompjuter ubaci na nekoliko turnira gde učestvuju ljudi. Doduše, u poslednje vreme se i to radi, ali i tu ima „kvaka". Čovek mora da odigra Indesetak partija da bi dobio rejting, a računaru se često meri samo jedan turnir. No, to je dovoljno da ga drugi proizvođači proćitaju. Krupna mana ne da se sakriti, a računari ih uvek imaju, što iskusni šahisti umeju vešto da iskoriste.

ZAJEDNIČKE MANE

Današnji šahovski programi se veoma razlikuju, ali ono po čemu su slični jesu mane! Neverovatno je da se one ponavljaju u seriji, kao da su klonirane! Čovek se jedva suzdrži da se ne upusti u akciju i napravi svoj program „samo da oni tamo na Zapadu vide kako se taj posao radi"! Recimo, treba namestiti tempo igre. Tu ima silesija opcija, ali se ispostavi da je one standardne opcije gotovo nemoguće podesiti. U šahu postoji 2.5 sata za 40 poteza (danas 2 sata), „blic" („cuger") – po 5 minuta za celu partiju, i „vikend" tempo – pola sata ili sat za celu partiju. I to je sve. To su fiksni „nivoi", uz koje bi trebalo dodati mogućnosti korekcija, kao i slobodu nameštanja vlastitih uslova. I svi bi bili zadovoljni! Uz to, obavezno bi se podešene opcije morale snimiti za sledeću seansu, da se čovek ne muči. Ali ne, kod većine programa to se mora ponovo svaki put!

Da bi se istestirala snaga programa, najbolje je odigrati sa njim poneku brzopoteznu partiju. Posle nekoliko partija videće da stalno gubite – zbog vremena! Onda se dosetite da provente da li vam računa vreme koje potroši na tehnički deo – odigravanje poteza. I naravno – računa! Tako postiže ogromnu vremensku prednost! On svoj potez „nacrta", a vi morate da primenite mehaniku! Neko opravdanje može da bude da se tako brani od nelegalnih poteza. I nema programa koji je to rešio na pravi način!

Nešto što je tekovina novijih programa, počev od *Chessmastera 3000*, jeste proračunavanje rejtinga. To postoji i kod *Zarkova*. Posle četiri partije imati gotovo velemajstorski rejting od 2469 jeste podvig, ali se postavlja pitanje valjanosti algoritma. Možda program ne bi trebalo da daje nikakvu informaciju o rejtingu dok se ne odigra 30 – 40 partije?

Sva ta silesija programa napravljena je na isti način – da se oslanja na dobar hardver. Oni ne misle – oni računaju! I odatle se izvod jedna konena osobina za praktičnu primenu u profesionalnom šahu. Naime, običan smrtnik ima racionalan pristup svakoj poziciji. On vrši selekciju poteza – kandidata prema određenim kriterijumima. Program ne! On daje sve što vidi! I zato može da predloži nešto što i najiskusnijem analitičaru promakne, a pokaže se vrlo jako! Na priloženom šahovskom dijagramu može se videti jedan dobar primer. Kad se uoče korisne osobine računara, one odmah nađu primenu. Zato su danas računari vrlo zahvalan pomoćnik doplenih igrača, pri traženju najjačeg poteza u igri, ali sigurno i u turnirskom šahu, pri analizi [illegible]nutih partija.



Siberian Chess: Jak shareware program

ZARKOV 2.60

Program *Zarkov* Džona Stenbeka (John Stanback), koji se uvek vrteo „tu negde", u pozadini, vratio se sa verzijom 2.60 iz 1992. godine. Zašto li se zove tako kako se zove, ostaje pitanje. Verovatno je autor u mladosti čitao „Flaša Gordona" i bio impresioniran [illegible]skom inteligencijom dr Zarkova. (Uzgred, kod Rusa ne postoji takvo prezime!) Bilo kako bilo, program je najzad počeo da „bije" i prave igrače. Grafika nije loša, ali nam se čini da je sve moglo biti i lepše urađeno, s obzirom da je tabla ogromna, preko celog ekrana. I sistem menija je za kritiku – u njega je uloženo taman toliko invencije koliko unose kipari-industrijalci u svoje programe! Tačka u kojoj treba kliknuti mišem uvek se besomučno traži, jer se njen položaj najtačnije definiše sa „tamo – negde – gore".

Nešto što je *Zarkov* uveo, a uvek je nedostajalo svima drugima, jeste opcija *Learn* – za učenje. Naime, većina programa je napravljena tako da u nekoj nepoznatoj poziciji uvek reaguje na isti način (ako se ne menjaju nikakvi parametri sistema – recimo raspoloživo vreme). Kad jednom tučete program, onda zapamtite partiju i samo ponavljajte poteze – to vam je uvek siguran poen! Neki su programeri pokušali da otklone tu slabost, uvodeći *Randomize* algoritam tako da mašina, kad ne može da se odluči između nekoliko nastavaka, bira onaj slučajni. Takve strategija ne pali uvek, jer će čovek zapamtiti i tih par varijacija i rezultat se opet zna. *Zarkov* svoje neuspehe pamti u poseban fajl, koji čak možete editovati i pojačati za njega, i tako zaobilazi sve hrid koje mu postavite. Samo je pitanje koliko je algoritam učenja inteligentan, te da li se kao rezultat učenja biraju bolji nastavci ili se samo varira. Najpre će biti da se postave neki limiti o kojima se zatim vodi računa. Još jedna pohvalna novost je „25. linija" na ekranu, ostavljena za komunikaciju sa programom. U nju možete ukucati komande, recimo predložiti remi (*draw*) ili predati (*resign*) partiju. Na žalost, ne reaguje na naredbu „*Predaj*". Vrlo zanimljivo. I pojačava utisak da igrate sa živim bićem! Kod novijih programa, takođe i kod *Zarkova*, postoji ugrađen takozvani *Opening books*. To su, u stvari, baze koje sadrže masu varijanti iz savremene teorije otvaranja, tako da lažu otvaranja svi savremeni programi igraju vrlo brzo i jako. Malo je prilike da iznenadite šahovski program nekim potezom u otvaranju, ako igrate iole standardno i pravilno. Ako malo skrenete ili igrate totalno iracionalno, nadigraće vas u otvaranju, jer su svakom od njih poznata opšte pravi-



Zarkov: Dijalog putem zelene linije



Chess Genius: I um i snaga!

... Recimo, ako igrate 1 h4 ili 1 a4, program neće puno razmišljati nego će zauzimati centar pešacima. Pa ipak, preporučujem ekstravagantu igru u otvaranju, ali u granicama legalnosti. Zanimljivo je da se transpozicijom, izvesnom izmenom u redosledu poteza u otvaranju, ne može zbuniti program. Otvaranja se prepoznaju samo po uvodnom nizu poteza, nego i po zaključnoj karakterističnoj poziciji varijanta pomoću *Hash* tabela. Nije dobro dokumentovano kako sve to radi, ali je pretpostavka da su tu sadržane najvažnije pozicije u otvaranju. Mnogi od programa rade daleko brže i igraju jače, bar u otvaranju, ukoliko imaju dovoljno memorije da se te tabele u nju smeste kao rezidentne.

Većina programa poseduje mnogo izrišne kozmetike, te naginju kičeraju. Na račun šminke, siguran sam da bi se mnoge ozbiljne opcije mogle usavršiti. Razvoj šahovskih programa dolazi do tačke kad se jasno treba povući granica u nameni. Jedni traba da koriste profesionalcima za trening, drugi amaterima za zabavu. Recimo, *Zarkov* ima pet nivoa gre, od novajlije (*Novice*) do stručnjaka (*Expert*). Po meni, to je sasvim nepotrebno. Svaki program trabe da igra samo na jednom nivou – punom snagom, pa ko izdrži. Još je gora mogućnost podešavanja raznih stilova. Na primer, napadački (*Aggressive*), pozicioni (*Cautious*) i univerzalni (*Balanced*).

Setimo se samo dobrog starog „Chessmastera 3000". Kod njega čak može da se nameša i individualni stil, pa da se s tim individuama napravi čitav turnir ... elo vam jedinstvene prilike da pobedite na turniru ispred Aljehina, Laskera i Kapablanke, a o Fišeru i Karpovu i da ne govorimo! Zaista divno, samo kada bi to bilo nešto više od zvečanje praznih slova! Ja sam pomalo eksperimentisao, ali bitne razlike u igri nisam video. Trebalo bi zaista mnogo vremena za testiranje, ali sumnjam da bi se i tada moglo pokazati drugačije. Zašto je to tako? Zato što je svaki program mogao biti konstruisan samo kao *individual*, kao jedan igrač, i samo tako i može da funkcioniše! Jedan Karpov može da odigra napadačku partiju, ali ne može imati čas ovakav, čas onakav stil. Praviti program koji u sebi ima ugrađen sukob ličnosti znači opteretiti ga u startu anarhičnom neefikasnošću!

NENADMAŠNI TAKTIČARI

Svi ovi programi imaju mogućnost da im se namesti dubina proračuna u poluhodovima (polupotez – potez samo jedne strane, ceo potez čine polupotezi obe strane). Mnogi od njih računaju vrlo duboko, devet polupoteza, pa i više. Sa ljudskog stanovišta ništa naročito, nije računati 4-5 poteza, ali programi uzimaju u obzir SVE mogućnosti! Tu situaciju lepo ilustruje klasična anegdota iz šahovske istorije. Na jednom jakom turniru, pre sedamdesetak godina, neko je vršio anketu među učesnicima koliko daleko proračunavaju poteze. Padali su razni odgovori, ta 5-6 poteza, te desetak, neko reče i do dvadeset, sve dok ne upitaše Čeha Retija, oca hipermodernog šaha, oslobođenog predrasuda u igri. Odgovor je bio „Svega jedan, najviše dva ... Ali jakih!"

Zbog tako detaljnog proračuna, programi su vrlo jaki u taktičkoj igri. Tako se opet vraćamo na hardver čija je usavršenost najviše uticala na popravu novih šahovskih programa, dok algoritmi manje više svi nalikuju jedan na drugog. Evidentna je stagnacija i samo se čeka neki novi impuls koji bi rešenje problema podigao za jednu stepenicu više. Zato bih preporučio i da ne pokušavate se njima da se nosite u taktičkoj igri, izuzev ako vam to nije jača strana, ili ako ne želite da vežbate taktiku. Svaki od novijih programa veoma je konstantan ako želite da usavršite svoju igru u brzopoteznim partijama, gde je veoma važno da činite što manje taktičkih previda. Svaki vaš sitan taktički propust biće momentalno kažnjen. To će vas naučiti pameti, te ćete ubuduće voditi više računa o svojim „drvcima"! Postoji jedna izuzetna situacija kada možete računar nadmašiti u taktici. To je slučaj ako mete dugotrajnu inicijativu ili atipican napad na kralja. Ne postoji nijedan program koji poseduje odbrambenu inventivnost na duže staze. Od opreznih pozicionih strategija, najuspešnije je igrati na jednaku ili nešto bolju poziciju, bez većih obaveza. Programi nisu sposobni da nađu pravi plan za igru u takvim slučajevima.

Način snimanja partija daleko je važniji za profesionalca nego za amatera. Amater obično poželi da sačuva partiju u kojoj nadigra računar, pa i da je odštampa na laseru da bi se pohvalio prijateljima („Da vidiš samo ... ale sam ga rasturio!"). S druge strane, profesionalac ima dovoljno vremena koje je odvojio za trening s računarom. On odigra masu partija, u kojoj se pojavi mnogo zanimljivih ideja i novih, neobičnih pozicija i novih rešenja – čiji je autor računar! Ponešto od toga može biti upotrabljeno u praksi. Zato je dobro imati ih negde spremljene, u obliku pogodnom za razgledanje. Svi programi mogu snimiti svoje partije. Nevolja je što je taj deo svuda urađen vrlo traljavo! Najnormalnije bi bilo očekivati da to bude urađeno u nekom od danas prisutnih standarda za šahovske baze partija, tako da se partije mogu lako „uvoziti". Na žalost, većina programa ima svoj format za snimanje, i to bi bilo podnošljivo kada bi svaka partija mogla da se snimi i u čistom ASCII obliku, jer šahovske baze poseduju takozvane ASCII *reader*-e. Ali, ne lezi vraže! Malo koji program zadovoljava takva kriterijuma. *Zarkov* puno ne mudruje, nego snima partije u ASCII obliku, u standardnoj, pristojno uređenoj šahovskoj notaciji. *Siberian Chess* ima svoj format i ništa drugo! *Chess Genius* ima i svoj format i ASCII, ali mu je ASCII prilično nesređen i onda morate praviti sopstvene makroe za prilagođavanje ASCII *reader*-ima. Uzgred, to je i jedina krupna mana ovog inače izvrsnog programa.

SIBERIAN CHESS 2.01

Program je nastao 1993. godine, autor je Eugen Naimov iz Novosibirska. Ono što impresionira kod ovog programa je njegova mala dužina – sva stane u 130K! Uz to, radi se o *shareware* verziji, pa ga zainteresovani mogu i sami pogledati. Potražite ga na SEZAMU! Ponešto je već rečeno o njemu, da dodamo još malo. On ima neobično grafičko rešenje ekrana, može se reći pre dobro nego loše. Smeta što je tabla pomerena sasvim u stranu ekrana. Sam izgled figura je vrlo originalan, privlačan, ali i pregledan. Poseduje najneophodnije opcije, ali je sve u rudimentarnom obliku – neracionalno je suviše tražiti od jednog *shareware* programa! Snaga igre vrlo je visoka za ovako „jeftin" program, no zaista je teško i nepotrebno rangirati programe po snazi. Svi su oni tu negde, a svako može da izabere ono što mu najviše treba. Predlažem da u savlađivanju protivnika započnete upravo od njega.

CHESS GENIUS 1.0

Datira iz 1992. godine, a autor je Richard Lang, čovek koji je prvi stvorio nešto vredno na PC-iju – „PSION CHESS". Poslednja verzija 2.0 navodno ima rejting kojeg se ne bi postideo ni velemajstor – 2.495. Od svih programa trenutno prisutnih na našem tržištu, nama dostupna verzija 1.0 zadaje igračima najviše problema. I zaista, ima neviđenu snagu koja nagoveštava da je „konačno rešenje" šahovskog pitanja barem što se računara tiče vrlo blizu. Sve je na njemu vrlo profesionalno. Od grafike, gde padaju u oči vrlo prijatne boje koje se mogu i birati, te pomalo jednostavan, ali nipošto ružan oblik figura, pa sve do snage igre koja se bez previše komplikacija, uz sitne intervencije, može znatno povećati. Sa opcijom *Permanent Brain* može se računaru dozvoliti da razmišlja o poziciji dok i vi razmišljate. Postoji nekoliko različitih vrsta biblioteka koje se mogu po želji menjati. Sve zavisi za koju svrhu program koristite. *Operator Time* opcija predviđena da vam obezbedi deo vremena za odigravanje poteza funkcioniše pomalo neobavezno. Umesto da zaustavi „štoperica" vašeg vremena u momentu kad dodirnete figuru, pa sve dok je ne ispustite na željeno polje, ona vam „refundira" utrošeno vreme u trenutku kada bi trebalo da izgubite na vreme!? Samo je to vrlo teško proveriti – hajde da poverujemo na reč! Jer, kad čovek počne često da gubi na vreme, teško mu je u to poverovati.

Zanimljiva je opcija *Pawn Structures*. Kaže se da, kad je uključena, program igra jače i logično je. U tom slučaju njegova igra je prilagođena pešačkoj strukturi. Jasno je da pravilna poziciona igra u najvećoj meri mora biti zasnovana na pešačkim obeležjima

Nastavak na strani 109

KAKO JE NASTAO IBM PC

Svako ko se bavi kompjuterima zna za IBM PC, računar koji je postavio mnoge (ako ne i sve) standarde u tadašnjem svetu ličnih računara i kojeg su kasnije nasledili IBM XT, AT, 386 i 486. Manje je poznata priča o njegovom nastanku. Evo je iz prve ruke, iz pera Dejvida Bredlija (*David Bradley*), člana tima koji je dizajnirao prvi IBM PC i tako, i neznajući, započeo novu stranicu u istoriji računara.

Avgusta 1980 godine završio sam posao na nečemu što je trebalo da bude IBM System/23 DataMaster. DataMaster je bio sistem baziran na procesoru Intel 8085 namenjen za izvršavanje poslovnih aplikacija napisanih u bejziku. Napisao sam nešto koda za kontrolu periferija i razmišljao o proširenju ka 8086 arhitekturi. Čuo sam za neku posebnu radnu grupu koja se interesovala za jeftino dizajniran sistem. Projekat je izgledao interesantno ali sam u tom času uživeo radeći na DataMasteru. No tada me pozvao moj šef i obavestio me da je IBM-ovo odeljenje u Boke Rejtonu (Boca Raton) dobilo zadatak da napravi jeftin sistem. Uprava je okupila tim inženjera i trebao im je neko da im napiše kontrolni kod. Tako sam ušao u tim u kojem je bilo još 11 ljudi.

U to vreme dizajn nove mašine postojao je samo u obliku nekoliko opisnih dijagrama. Godinu dana kasnije, 12 avgusta 1981, predstavili smo javnosti IBM Personal Computer. Evo kako smo odlučivali i šta nas je vodilo prilikom dizajna i uobličavanja IBM PC-ja.

SKROMAN POČETAK

Projekat DataMaster započet je februara 1978. Mnogi elementi njegovog dizajna – tastature, na primer – upotrebljeni su kasnije na IBM PC-ju. Istovremeno, rad na razvoju DataMaster-a proizveo je grupu inženjera verziranih u oblasti dizajna ličnih računara. DataMaster je oblikovan kao jedna fizička celina. Jedan komplet obuhvatao je tastaturu, 12-inčni monohromatski ekran i dve flopi-disk jedinice od 8 inča. Takvo rešenje bilo je pogodno za poslovno okruženje kome je bio namenjen System/23, ali nije imalo fleksibilnost jednog ličnog računara. DataMasterov 8-bitni CPU 8085 bio je veoma bliski rođak procesoru 8080 i imao je isti skup instrukcija. Tokom rada tima na razvoju DataMastera iskoračili smo izvan ograničenja adresnog prostora od 64 kB i morali smo izmisliti jedan spoljni mehanizam da bismo adresirali najmanje 256 kB memorije. Dizajnirajući taj sistem dobro smo upoznali Intelovu arhitekturu i prateće čipova. Sistem DataMaster je, kao i mnogi lični računari tog vremena, imao ugrađen BASIC interpreter. Tokom razvoja DataMastera „priklonili" smo naš BASIC onom koji je koristio računar IBM System/34, što je odložilo završetak DataMastera za godinu dana. Iz tog iskustva smo naučili šta treba da učinimo da se proizvod što brže iznese na tržište: prvo, trebalo je da upotrebimo neki već postojeći BASIC, i drugo, trebalo je da „podmažemo" IBM-ov razvojni proces. Upravo je primena ovog iskustva jedan od razloga uspeha IBM PC-ja. Do Microsofta smo otišli po BASIC interpreter a sproveli smo i naš „ubrzani razvojni proces", koji je između ostalog obuhvatao upotrebu izvršnih direktora za ubeđivanje ostalog dela korporacije da smo na pravom putu.

PRVI POGLED

Prilikom krajnje dorade originalnog PC dizajna zamenili smo mesta izvoru napajanja i karticama za proširenje, omogućujući tako adapterima da se protegnu duž celog kućišta. Tim potezom je prekidač za uključivanje računara dospeo nazad u ugao. U osnovni dizajn smo uvrstili par flopi disk jedinica od 5,25 inča, horizontalnu sistemsku ploču i pet utičnica (slotova) za proširenje. Zbog iskustva sa DataMasterom tastaturu i ekran smo odvojili od samog računara. Osnovna logika je bila da kartice za proširenje svakom sistemu daju jedinstveni „šarm". Kućište je sa svojim dimenzijama moralo da se smesti na standardan sto, a da još ostane mesta i za tastaturu. Prostor u kućištu preostao posle smeštanja izvora za napajanje i disk jedinica odredio je veličinu sistemske ploče i kartica za proširenje.

Sistem dizajniran u početku bio je dosta drugačiji od onog koji smo na kraju predstavili. Sve sisteme smo isporučivali sa najmanje 16 kB RAM-a na sistemskoj ploči. Svi sistemi su imali 40 kB ROM-a koji se sastojao od BASIC interpretera (32 kB) i BIOS-a (8 kB). I u prvobitno zamišljenoj mašini, kao i u krajnjoj, nalazilo se 5 utičnica za proširenje. Na kraju smo zapravo ugradili većinu karakteristika originalnog DataMastera, sa izuzetkom 8-inčnih disk jedinica, dok smo priključak za štampač izmenili u paralelni koji je bio industrijski standard.

Najveći uticaj na dizajn sistema imao je tek usvojen FCC Class B atest kućnih elektronskih uređaja. U tim ranim godinama „ojačati" sistem da spreči elektromagnetske emisije bio je težak zadatak, ponekad više umetnost nego nauka. Trebalo nam je mnogo časova testiranja da bi naš sistem zadovoljio FCC kriterijume.

Razvijajući DataMastera i iz iskustva drugih, naučili smo da čak ni kompanija veličine jednog IBM-a nije u stanju da proizvede sav hardver i softver potreban da jedan lični računar uspe na tržištu. Odlučili smo da od samog početka objavljujemo sve podatke u vezi sa svim hardverskim i softverskim interfejsima. Svakom ko je razvijao neki adapter ili program za IBM PC pružali smo sve raspoložive informacije. Kompilacijom svih sistemskih specifikacija korišćenih tokom razvoja sistema i njegovog testiranja nastao je „Technical Reference Manual".

ZAŠTO 8088?

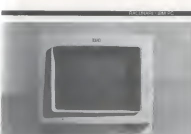
Postojali su brojni razlozi zbog kojih smo odabrali Intel 8088 za centralni procesor IBM PC-ja. 1) Ograničenje na adresiranje najviše 64 kB memorije moralo je biti prevaziđeno, takav zahtev je povlačio za sobom upotrebu 16-bitnog procesora. 2) Potrebno je bilo da procesor i njegovi prateći sklopovi budu odmah raspoloživi. Nije bilo vremena da se čeka na razvoj novih LSI čipova, jer su nam odmah bile potrebne odgovarajuće količine. 3) Nismo mogli da sebi pustimo vreme za učenje novih stvari, morali smo upotrebiti tehnologiju koja nam je bila bliska. Takođe nam je trebao i raznovrstan skup pratećih čipova – želeli smo da sistem ima DMA kontroler, kontroler prekida, tajmere i paralelne portove. 4) Za procesor su morali biti na raspolaganju i operativni sistem i aplikativni softver.

Izbor smo suzili na Intel 8086 ili 8088. Inženjeri u Boke Rejtonu su dobro poznavali ove procesore i njihove prateće čipove. Za programere, 8086/8088 arhitektura je direktna nadgradnja 8080 arhitekture. Microsoft je svoj BASIC interpreter već bio preneo sa 8080 na 8086. Postojao je takođe i DOS koji se izvršavao na 8086. Proširenje adresnog prostora je bio naš primarni cilj. Izabrali smo na kraju 8088 zbog njegove 8-bitne magistrale podataka (*data bus*). Uža magistrala je uštedela novac kod RAM-a, ROM-a i logičkih kola. Bonus koji smo dobili izborom ovog Intelovog procesora bio je numerički koprocesor 8087. 8087 je bio brzi pomoćnik 8088-ici pri računanju sa pokretnim zarezom. Međutim, pošto 8087 još nije bio na raspolaganju kada je najavljen PC, jednostavno smo ostavili na ploči jedno prazno podnožje za koje nismo saopštili namenu.

OD 16 kB PA NAVIŠE

IBM PC je kupcima ponuđen sa 16 kB RAM-a koji su se mogli proširiti do 64 kB na samoj ploči. Upotrebili smo DRAM čipove kapaciteta 16 kB, jer su oni tada bili standardan proizvod. Projektovali smo i dve kartice za proširenje memorije, jednu od 32 kB i drugu od 64 kB. Sa maksimalno konfigurisanom sistemskom pločom i tri kartice sa po 64 kB, dobijali ste 256 kB memorije, količinu koja je u to vreme bila impresivna za jedan lični računar. Uradili smo i nešto bez presedana kod ličnih računara – obezbedili smo proveru ispravnosti memorije. Smatrali smo da je veoma važno otkriti greške koje bi mogle biti uzrokovane neispravnom memorijom i sprečiti „pokvarene" podatke da se dalje prostiru sistemom. Verovali smo da je bolje zaustaviti mašinu nego nastaviti rad sa greškama. Kada je neki aplikativni program želeo da izmeni način na koji se obrađivala greška u paritetu, on je to činio jednostavnom promenom obrađivača (*handler*) nemaskirenog prekida.

Zašto je DOS na kraju ostao ograničen na 640 kB? Odgovor je iznenađujuće jednostavan. Procesor 8088 može da adresira 1 megabajt. Gornjih 128 kB smo rezervisali za ROM na sistemskoj ploči. Želeli smo da video memorija (ekranskih adaptera) bude u adresnom prostoru procesora, tako da smo i za nju rezervisali 128 kB memorije. I na kraju 128 kB memorije smo rezervisali za ROM ili RAM na drugim adapterskim karticama. U to vreme smo bili mišljenja da je takva raspodela prilično velikodušna. Kada se oduzmu rezervisane oblasti – ostaje 640 kB. S obzirom da je kapacitet RAM-a originalnog IBM PC-ja bio 256 kB, dok je za sisteme tog vremena bilo uobičajeno da imaju 64 kB, mi smo sa našim rešenjem bili veoma zadovoljni. Mada 640 kB jeste postalo ograničavajuće, isto tako su to postala i druga rezervisana područja. Smatram da nije bilo boljeg načina da se organizuje memorija, a koji bi doneo neko bitnije poboljšanje.



PODRŠKA PROCESORU

Pomoćni čipovi (*support chips*) su povećali funkcionalnost IBM PC-ju. DMA je ubrzavao U/I (ulazno-izlazne) operacije oslobađanjem procesora od prenosa podataka između memorije i U/I uređaja. Želeli smo i da sistem bude u stanju da izvršava više operacija istovremeno. Na primer, da korisnik kuca na tastaturi dok istovremeno disk radi sa podacima. Takve mogućnost povećala je „udobnost" i upotrebljivost sistema. DMA smo takodje iskoristili i za osvežavanje memorije. Mogućnost kucanja unapred zahtevala je i tastaturne prekide (*keyboard interrupts*).

Upotreba kontrolera prekida sa 8 nivoa oslobodila je procesor od stalnog proveravanja U/I operacija. Mada hardver i prateći softver koji je isporučivan uz mašinu nisu u potpunosti iskorištavali pogodnosti DMA i mogućnosti prekida (interapta), ta smo funkcije ugradili za druge koji će ih iskoristiti. Na primer, BIOS za serijski port nije koristio mogućnosti prekida, ali su ih zato koristile serijske komunikacije ugrađene u *Advanced BASIC*. Serijski prenos i prijem mogao sa obavljati pod kontrolom prekida dok se istovremeno izvršavao BASIC program. Trokanalni tajmer/brojač omogućavao je IBM PC-ju da pokazuje tačno vreme bar dotle dok je računar bio uključen. Uz to, brojač je mogao biti upotrebljen za mnoge vremenske funkcije sistema – na primer za kod koji je u BIOS-u čitao podatke sa kasetofona. Taj kod je koristio tajmer da odredi dužinu ciklusa bita i na taj način odluči da li se radi o snimljenoj jedinici ili nuli.

BOJA I TEKST

IBM PC-ju smo namenili dve različite uloge: da bude poslovni i da bude kućni računar. U cilju podrške toj višestranosti razvili smo posebne videoadaptere: CGA za kuću, a MDA (*Monochrome Display Adapter*) za posao. MDA se bazirao na režimu DataMastera koji je bio monohromatski i samo tekstualni. DataMaster je za generisanje prikaza na ekranu upotrebljavao CRT kontroler Intel 8275. Kod ovog kontrolera su postojale dva problema: bio je ograničen na 7 bita po karakteru tako da je mogao prikazati samo 128 različitih znakova, i nije mogao postaviti atribute znakova a da ne žrtvuje poziciju znaka specifikaciji atributa. MDA je imao jedan režim: tekstualni 25x80 znakova. Nismo iskoristili rešenje sa DataMastera jer smo više voleli da uradimo sopstveno rešenje – uz pomoć Motorole 6845 kao video tajmera, i dodatne memorije na kartici za smeštanje znakova i atributa. Takvo rešenje nam je omogućilo da definišemo 256 znakova i da po potrebi definišemo atribute za svaku poziciju na ekranu.

MDA je bila prva višefunkcijska ploča razvijena za IBM PC. Posedovala je jedan paralelni port za vezu sa štampačem. Trebalo nam je najmanje 256 znakova da bismo podržali glavne jezike, jer ipak slovo I u IBM znači „internacionalan". Sa DataMastera smo preneli prilično dobar i upotrebljiv skup. Takodje smo želeli da u našem skupu znakova imamo i „poslovnu grafiku" (tj. linije i uglove upotrebljive za pravljenje okvira). Preostala mesta popunili smo nekim oznakama iz oblasti obrade teksta, grčkim slovima i nekim matematičkim simbolima.

Što se tiče CGA, želeli smo da radi i na kućnom televizoru kao i na monitoru TV frekvencije. Takav stav je u velikoj meri uslovio sve ostale bitne stvari kao što je na primer broj tačaka na ekranu. Dva od nekoliko ekranskih režima rada, tekstualni 25x40 znakova i grafički 320x200 tačaka od kojih se svaka mogla posebno adresirati, bili su posledica ograničenja kućnog TV-a. Preostala dva režima, tekstualni 25x80 znakova i grafički 640x200 tačaka, za svoj red su zahtevali monitor.

Mogućnosti prikazivanja boja bile su određene količinom memorije koju smo sebi mogli priuštiti na matičnoj ploči – 16 kB RAM-a. Ovolika memorija obezbeđivala je 4 boje za grafiku srednje rezolucije (320x200) i 2 boje pri visokoj rezoluciji. Zbog organizacije memorije sa 16 kB po 8 bita, pojavio se jedan od problema sa CGA: u tekstualnom režimu sa 80 znakova, adapter je morao da prebacuje 160 bajtova po liniji – 80 za znakove i 80 za atribute, ne ostavljajući pri tom vremena procesoru da čita ili piše na ekran. Svaki program koji bi išao direktno na ekransku memoriju ubrzo bi ispunio ekran „snegom". Rutine BIOS-a za čitanje i pisanje čekale su na povratak horizontalnog signala da bi izvršile pristup memoriji. BIOS rutine za skrolovanje, koja je morala da prebacuje velike količine podataka, jednostavno je isključivala ekran, vršila premeštanje, a zatim uključivala ekran. Zato je pri skrolovanju ekrana bilo primetno treperenje, ali je to bilo manje zlo od bljeskanja koje bi se inače dobilo na ekranu.

KASETE PA DISKETE

Pošto smo IBM PC namenili i za kućnu upotrebu, uvrstili smo u njegove priključke i port za povezivanje kasetofona kao spoljašnjeg medija za čuvanje podataka. Sistem nije imao ugrađene disk jedinice, imao je 16 kB RAM-a i kasetni BASIC (*Cassete BASIC*) u ROM-u. Sa ovakvom konfiguracijom mogli ste učitavati i izvršavati BASIC programe pomoću običnog kasetofona. Pa ipak, dizajn sistema smo optimizovali i za poslovnu upotrebu, sa prostorom za dve flopi-disk jedinice i 64 kB RAM-a na osnovnoj ploči. Kasnije se pokazalo da su se sistemi sa disk jedinicama daleko bolje prodavali nego oni bez njih, pa je gotovo svako ko je kupio sistem baziran na kasetofonu kasnije dokupio i disk jedinicu. Port za kasetofon je zadržan u PC-ju zbog kompatibilnosti, a uklonili smo ga kad je nastao XT.

ARHITEKTURA MAGISTRALE

Arhitektura magistrale (*bus*) IBM PC-ja nastala je iz dva izvora: magistrale DataMastera i novih potreba procesora 8088. Želeli smo da nova magistrala ostane što sličnija DataMaster-ovoj, pošto smo za ovu drugu već imali nekoliko adapterskih kartica. Sličnost magistrale činila je adaptaciju veoma jednostavnom – trebalo je samo preoblikovati štampu na karticama. 62-pinski konektor takođe je preuzet sa DataMastera. Proširenja za 8088 su bila očigledna – nekoliko ekstra adresnih linija. DataMaster je koristio jednak kontroler prekida i sličan DMA kontroler, tako da su se ti signali već nalazili na DataMaster magistrali. Na kraju je promenjeno samo pet originalno predloženih signalnih linija.

Jedna stvar koju nismo predvideli u vreme kada smo definisali sistemsku magistralu bila je lavina adapterskih kartica za PC. Mada se pokazalo da je takva raznovrsnost bila dobra, jer je pružala kupcima širok izbor mogućnosti, mi nismo obezbedili dovoljno U/I adresa za sve te kartice koje su razvijene. Originalne kartice za proširenje su dekodirale samo 10-bitnu adresu, a mnoge kartice su koristile 8 ili 16 adresa. Zbog toga je gotovo svaka kartica imala skup prekidača za izbor U/I adrese na kojoj će raditi i liniju prekida koju da koristi na datom sistemu. Takvi DIP prekidači su primoravali korisnika da svaki put pre instaliranja nove kartice pregleda svu dokumentaciju. Rešenje za ovo je bio programabilni metod izbora uređaja, što je bio jedan od razloga za prelazak na Micro Channel arhitekturu kod PS/2 familije.

JOŠ SAMO DA BUDE PARA...

Za promociju svog softverskog proizvodnog programa mnoge domaće firme biraju program za obradu ličnih dohodaka, plata ili zarada – već kako trenutak nalaže da se ovakav program nazove. Pred nama je najnoviji ovakav proizvod beogradske firme "Mega computer engineering".

Zoran Živetić

Program za obračun plate je po mnogo čemu specifičan i predstavlja dobar test sposobnosti firme da proizvede i održava program iz oblasti klasičnog knjigovodstva. Novi poreski zakon i prateći zakoni koji su stupili na snagu januara 1991. godine su davali nadu da će problem obračuna plata biti skinut sa dnevnog reda jer su konačno uveli jednostavan i jasan sistem obračuna. Na žalost radost je bila kratkog veka. Princip "radnik dobija bruto iznos i iz njega namiruje svoje poreske i druge obaveze" gotovo da nikade nije ni prihvaćen u našoj sredini. Mnoge firme su nastavile da obračun izvode po principu "ako radniku treba isplatiti toliko i toliko, daj da izračunamo koliki je to bruto da bismo zadovoljili zakonsku formu, pa će primenom obračuna unazad sve opet biti po starom". Stvar se vremenom dodatno komplikovala raznim izmenama zakona i novim uredbama koje su parcijalno regulisale pojedine segmente obračuna. Tako da su nastajale apsurdne situacije. Tako je u jednom trenutku, nakon izmene poreske skale postojao određen opseg neto iznosa koji je nemoguće isplatiti – preračunom ovog iznosa na bruto i zatim primenom obračuna unazad ne može se dobiti ponovo taj iznos neto zarade. Danas je stanje nešto bolje, ali i dalje postoje specifične situacije za koje, bar koliko je autoru ovog prikaza poznato, niko precizno ne zna kako bi obračun, ukoliko se zakon striktno primenjuje, trebalo da izgleda. S obzirom na visinu zarada, ovaj problem je više teorijske prirode, jer se praktično svodi na gotovo zanemarljive razlike u iznosima poreza ili doprinosa. Ali, ono što se u praksi "ručno" lako razreši, pred programera koji mora da izvrši formalizaciju algoritma nameće ozbiljne probleme.

Da bi čitaocu neupućenom u ovu materiju bilo do kraja jasno kakve probleme mora da rešava program za obračun plata, reći ćemo još samo da se i izračunavanje osnovnog neto iznosa koji radnik treba da primi, s obzirom da je taj deo uglavnom interna stvar svake firme, razlikuje i ima mnogo specifičnosti. Ako se na ovo doda i činjenica da veće firme često za pojedine svoje segmente primenjuju različite načine određivanja neto iznosa plate, da potrebnu dokumentaciju za isplatu formiraju na nivou cele firme, a da povremeno zahtevaju i analitičke izveštaje po segmentima grupisanim prema različitim kriterijumima, jasno je da nešto što bi se zvalo "potpuna automatizacija obračuna plata" i što bi važilo za sve firme ne postoji. Svaki program zato pokriva manji ili veći segment problema. Da vidimo konačno kako se sa svim ovim problemima nosi program *Zarade 2.1*.

INSTALACIJA

Program *Zarade* verzija 2.1 se isporučuje na tri HD diskete uz skromno uputstvo malog formata na dvadesetak stranica. Instalacija je potpuno automatizovana – korisniku ostaje da izabere samo oznaku tvrdog diska na koji program želi da instalira i za manje od pet minuta na tom disku će se naći direktorijum MEGA i njegov poddirektorijum *Zarade* u kome se nalaze program i podaci. Na disku će se naći i poddirektorijum VODIC u kojem se nalaze podaci za "samo-izvođenje" demo verzije programa koja korisniku, bez njegove intervencije, prikazuje osnovne mogućnosti programa.

Minimalna konfiguracija na kojoj program radi je 386 računar sa bar 2MB RAM memorije od čega treba da ima bar 550KB DOS i 1MB XMS memorije slobodno. U uputstvu se pominje i mrežna varijanta koja zahteva posebnu instalaciju na serveru mreže, nakon čega zahteve radne stanice može da ispuni i 286 računar bez XMS memorije.

RAD SA PROGRAMOM

Nakon startovanja programa *Zarade* i upisa lozinke, ekran koji se pojavljuje odmah ukazuje da se radi o, za naše prilike, nekonvencionalnom proizvodu. Umesto prikaza menija sa osnovnim opcijama programa na ekranu odmah imamo tri prozora koja nude pristup pojedinim segmentima podataka potrebnim za obračun. U jednom se vidi struktura preduzeća, u drugom su podaci o radnicima, dok su u trećem podaci za izvođenje obračuna. U svakom prozoru na dnu se nalaze oznake operacija prikazane kao tasteri (ili dugmad, zapravo kao tekst varijanta *Windows* kontrolnih dugmadi). Tako se prelaskom (tasterom TAB) na dugme **Dodaj** i pritiskom na **Enter** dodaje novi podatak. Moguće je i brže – pritiskom na **Alt** taster i slovo prikazano naglašenom bojom operacija se izvodi u jednom potezu.

Na prvoj liniji ekrana nalazi se klasičan *pull-down* meni u kojem su sve opcije – najveći broj njih zapravo otvara nove prozore u kojima se nalaze dodatni podaci potrebni za obračun – spisak kreditora sa žiro-računima, spisak banaka, lista samodoprinosa, sindikati, itd. Za razliku od već pomenuta tri osnovna prozora koji mogu biti stalno prikazani na ekranu i između kojih se možete slobodno kretati tasterom Ctrl+TAB (u krug, ili brže, sa Alt+/broj_prozora), aktiviranje nekog od dodatnih prozora zahteva da rad sa njim završite i zatvorite ga kako biste mogli da nastavite rad.

Priprema podataka za obračun počinje definisanjem strukture firme. U prozoru "Struktura preduzeća" unosi se struktura u formi razgranatog stabla – počnete sa nazivom firme, zatim "uđete" u njegovu strukturu i dodajete nove segmente ili nove grane do nivoa i količina koje su neograničene. Kroz ovu strukturu se krećete poput kretanja kroz direktorijume diska – korak u nazad prikazan je standardnom oznakom "..".

Izborom neke grane i prelaskom u prozor "Radnici" unosite radnike koji se u tom odeljenju nalaze. Radnici se mogu nalaziti na svakom nivou bez obzira da li taj nivo organizacije ima i svoje podnivoe. Za svakog radnika predviđen je unos imena, matičnog broja, broja izdržavanih lica, broja bodova, radnog staža i datuma uvećanja radnog staža. Isplata predviđa mogućnost da radnik dobija platu preko dve banke i na blagajni – za svako od isplatnih mesta unosi se procenat koji se na tom mestu isplaćuje dok se za banke unosi i broj tekućeg računa. Time još nije definisano o kojoj se banci radi – posebnom opcijom se poziva spisak banaka i iz njega banka "vezuje" za jedno od isplatnih mesta izabranog radnika.

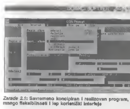
Ovo je inače način na koji se ostvaruju stale veze. Ako radnik uplaćuje neki samodoprinos, iz liste samodoprinosa se jedan od njih vezuje za radnika. Ako uplaćuje sindikalnu članarinu, u listi se bira sindikat i zatim vezuje za radnika. U oba slučaja iznos se određuje procentom iz zarade. Obezbeđena je automatska praćenje kredita. Prvo se bira kreditor iz liste, zatim se zadaje iznos i broj rata i program automatski formira listu vraćanja kredita po mesecima. Lista se može naknadno ručno menjati.

PRAĆENJE ZAKONSKIH IZMENA

Obračun se može izvršiti kao akontativni obračun ili konačni obračun. U drugom slučaju za obračun se vezuju akontacije koje su već isplaćene čime se dobija zakonom propisan kumulativni obračun. Može se birati i princip obračuna – neto na bruto ili bruto na neto.

Novi obračun se dodaje jednostavnim ubacivanjem u listu postojećih obračuna nakon čega sledi unos zajedničkih podataka (faktori vrednovanja pojedinih vrsta rada kao što su redovan rad, rad praznikom, plaćeno odsustvo, bolovanje do i preko 60 dana itd) i inicijalizacija obračuna. Inicijalizacija se može obaviti na nivou jednog odeljenja u strukturi preduzeća ili na nivou tekuće i svih podgrana. Zatim se kretanjem kroz listu radnika vrši pregled i mesečna korekcija za one radnike koji u tekućoj isplati imaju određene specifičnosti (manji fond časova, bolovanje itd). Ova procedura se izvodi tako što se posebnim tasterom dobija prozor sa svim podacima o isplati – zajedničkim podacima, podacima za radnika i u desnom delu podacima proisteklim iz obračuna. Ova poslednja površina sadrži sve međurezultate obračuna, tako da se tok primenjenog obračuna lako može pratiti i kontrolisati. Na pojedinim poljima ove liste se vrše ručne korekcije prema potrebama tekućeg obračuna za izabranog radnika. Nakon svake korekcije njeni efekti na obračun se odmah vide.

Ono što obično korisnike na samom početku interesuje je garancija koju proizvođač daje da će pratiti buduće zakonske promene. Dizajn ovog programa u velikoj meri predviđa jednostavnu mogućnost praćenja zakonskih izmena. U delu programa koji je posvećen naprednijim korisnicima otkriva se da se zapravo obračun odvija tako što se vrednost pojedinih polja izračunava preko formula u kojima druga polja učestvuju kao promenljive. Formule dozvoljavaju ne samo osnovne aritmetičke operacije već i poznatu C-ovsku konstrukciju (logički izraz) ? (uradi ako je tačan) : (uradi ako nije tačan). Korisnici će ipak imati malo direktne koristi od ove fleksibilnosti. Najčešće izmene zakona do



Zarade 2.1: **Savremeno koncipiran i realizovan program, koji nudi mnogo fleksibilnosti i lep korisnički interfejs**

sada su se odnosile na način računanja poreza i doprinosa a formule koje se tiču ovog dela su veoma komplikovane (u smislu složenosti izraza, matematički svakako ne) čak i za iskusnijeg korisnika pogotovo što je način njihovog unosa izuzetno nepregleden (jedna linija čiji se sadržaj samo delimično vidi). Moguće su i situacije u kojima izmena formula neće pomoći – ukoliko se recimo poreske skale iskomplikuje tada će biti potrebno uvesti nova polja u obračun (u programu su ostavljena i neka dodatna za sada neiskorišćena.) Ipak, iako smo ubeđeni da se ni jedan od korisnika neće usuditi da "čačka" po ovom delu, verujem da će ipak umeti da cene ovu fleksibilnost programa jer im pruža određenu garanciju da firmi neće trebati puno vremena da program prilagodi novonastaloj situaciji.

Fleksibilnost od koje korisnici mogu imati više koristi je mogućnost da direktno utiču na izgled štampanih izveštaja. Posebnom opcijom programa se za svaki od izveštaja dobija u obliku teksta "formular" po kome se izveštaj štampa. Program se praktično ponaša kao generator izveštaja kome se jednostavno prosleđuje opis po kome treba da obavi posao.

Program je opremljenim i nizom opcija za podešavanje – od znaka pozadine, preko boja svakog od elemenata korisničkog interfejsa do izlaznog uređaja, dužine strane i margine kod štampanja. Ugrađen je i mali kalkulator koji se može pozvati tasterom Alt+Q u svakom trenutku rada. Kalkulator, na žalost, nema mogućnost da se dobijeni rezultat prenese u polje

VISOKI KRITERIJUMI

Odlična koncepcija i realizacija ovog programa nameću da na njegovo ocenjivanje primenimo veoma visoke kriterijume. U tom smislu imamo i nekoliko zamerki od kojih ni jedna ne utiče na upotrebljivost programa, ali svakako mogu da utiču na ocenu koju program mogu dati krajnji korisnici. Sve zamerke zapravo proizilaze iz pomalo nejasnog balansiranja između želja da program bude namenjen potpunim početnicima ali i iskusnijim korisnicima. S obzirom da je broj ovih prvih daleko veći smatramo da su potrebna određena sitnija "peglanja" korisničkog interfejsa. Na primer, kada se pojavi prozor sa nekim obaveštenjem u dnu ekrana stoji informacija da se prozor uklanja pritiskom na "bilo koji taster" – prozor se ipak može ukloniti samo pritiskom na taster ESC. Pritisak na taster ESC u prozoru sa obračunima daje poruku da ste zašli u deo programa koji ništa ne radi i da za nastavak treba pritisnuti Ctrl+TAB ili F10 – pritisak na ove tastere ne daje nikakav efekta jer prozor treba prvo ukloniti tasterom ESC pa tek potom primeniti jedan od ove dve.

Svaki od prozora se može može pomeriti ukoliko zaklanja neki važan podatak iz drugih prozora ali se nova pozicija pamti samo za tri osnovna prozora – pomerite neki drugi prozor, izađite iz njega i kada se ponovo vratite prozor je ponovo na starom mestu. Program u nekim situacijama izdaje upozorenja koja su za početnika potpuno nejasna (poput SQL je dojavio internu grešku...) iako nakon toga normalno nastavlja sa radom. Verujemo i da bi firma i sebi i korisnicima uštedela nešto vremena i nerviranja kada bi u opciju za izmenu formule obračuna ugradila i komandu "poništi izmene i vrati originalne formule".

Uputstvo, ugrađenu pomoć pa i samoizvršavajući ČMDULĆVodičČMDNMĆ takođe pate od sličnih simptoma. Informacije su prilično nedorečene i uglavnom opšte prirode ("u ovoj masci se unosi to i to") što je inače jasno, dok o specifičnosti pojedinih polja nema reči. ČMDULĆVodičČMDNMĆ se zapravo mnogo više bavi interfejsom i njegovim mogućnostima nego samim programom, tako da korisnik, pre svega zainteresovan za obračun plata iz njega neće steći pravu sliku o mogućnostima programa.

Ipak jedini stvarno ozbiljan nedostatak koji se najlakše može i otkloniti je nepostojanje ugrađene i jasne procedure za pravljenje zaštitnih kopija podataka – backup. Ovde su korisnici upućeni na firmu da im ona na licu mesta organizuje backup proceduru. Jednostavan backup celog direktorija, za čim bi prosečan korisnik mogao posegnuti, znači i stalno nepotrebno snimanje i programskih datoteka koje svojom veličinom prevazilaze prostor koji podaci zauzimaju. Možda bi bilo dovoljno da se bar odvoje direktorijumi za podatke i izvršni kod što bi inače bilo pozdravljeno od korisnika koji ozbiljno koriste lokalnu mrežu.

Program *Zarade 2.1* predstavlja savremeno koncipiran i realizovan program (pisan je u *Watcom C++* i oslanja se na *Watcom SQL* mašinu) koji nudi mnogo fleksibilnosti i lep (verujemo za mnoge) korisnički interfejs. Njegovo korišćenje je jednostavno i verujemo da će i korisnici navikli na krutost uobičajenih domaćih interfejsa lako prihvatiti pogodnosti koje nudi. Nadamo se da će Mega "pakovanje" ovog programa uskladi sa njegovom vrednošću što zahteve da se uputstvom i vodičem ozbiljnije pozabavi.

Ukoliko vaša firma spada u veće u kojima obračun, zbog velikog broja radnika, zahteva istovremenu pripremu podataka na više radnih stanica, obavezno se raspitajte za mogućnosti mrežne verzije koja se pominje u uputstvu – ona nam nije bila dostupna za ovaj prikaz.

Cena: 700 ndin

Korisna adresa

Mega computer eng[illegible]ng
11000 Beograd, Bulevar Revolucije 316
Telefoni: 412 779 i 421 993

Nastavak sa strane 79

KONTROLA TASTATURE

Tastatura IBM PC-ja preuzeta je, kao što je ranije rečeno, od DataMastera. Izmenili smo natpise na nekim tasterima da bismo istakli opštu namenu PC-ja (na primer, labelu HELP smo zamenili sa F4). Kako je tastatura DataMastera bila u sklopu sistemskog kućišta, upotrebljavala je paralelnu vezu sa sistemskom pločom. Kod IBM PC-ja smo tastaturu odvojili od kućišta i povezali je sa sistemom preko serijskog porta i dvometarskog spiralnog kabla.

Jedno od pitanja koje su nam često postavljali u vezi sa IBM PC-jem je bilo zašto smo izabrali baš Ctrl-Alt-Del kombinaciju za resetovanje računara. Odabrali smo baš ove tastere za „vrući restart" iz više razloga. Kao prvo, teško ih je slučajno zajedno aktivirati, pošto su tasteri široko razmaknuti. Vrući restart je smeštanjem specijalnog znaka na određenu memorijsku lokaciju preskakao testiranje memorije (u sklopu autotestiranja prilikom uključenja) tako da je trajao kraće nego isključivanje i ponovno uključivanje mašine. Međutim, nevolje sa Ctrl Alt Del sekvencom je u tome što je za uspešno resetovanje potrebno da sistem bude u stanju da očitava tastaturu. Ali ako je neki program kompletno pobrljavio i onemogućio interapte, reset neće funkcionisati.

Na originalnoj tastaturi su vidljive još dve stvari: položaj levog *Shift* i *Enter* tastera, i nepostojanje indikatora za *Caps Lock*, *Num Lock* i *Scroll Lock*. Tasteri su postavljeni tako kako jesu jer smo želeli da podržimo međunarodnu tastaturu, koje ima nekoliko dodatnih tastera. Poboljšanje smo izveli na IBM AT tastaturi koja je imala indikator i za stanje *Shift* tastera.

SISTEMSKI SOFTVER U ROM-u

Namera nam je bila da PC-jev BIOS služi kao bafer između hardvera i programera. Neke stvari koje BIOS obavlja (na primer, slanje znaka štampaču) su veoma jednostavne. Druge operacije (na primer, dekodiranje kodova očitanih sa tastature u željene znakove) su vrlo složene. Želeli smo da programeru omogućimo da se usredsredi na sam problem a ne na hardver.

Drugi važan deo IBM PC-ja je bila njegova dijagnostika. Autotest pri svakom uključenju računara nervirao je neke korisnike jer je trebalo više vremena da sistem postane spreman za rad. Ali mi smo bili mišljenja da je važno testirati sistem pre pokretanja neke aplikacije, rezonujući da je bolje otkriti hardverski problem pre nego što on prouzrokuje greške u podacima.

INDUSTRIJSKA REVOLUCIJA

Kada smo započinjali razvoj IBM PC-ja, potcenili smo njegove tržišne potencijale. Kompanija je tada procenila da će prodati ukupno oko 250 000 ovih računara u periodu od 5 godina. Ali, kako se pokazalo, bio je potrebno samo nekoliko meseci da se napravi i proda otprilike taj broj računara.

Od tada je IBM tržištu isporučio milione PC-ja. Procene prodaje su možda (?!) i bile pogrešne, ali je zato sam proizvod bio pun pogodak. Veoma sam srećan što sam bio deo svega toga.

David J. Bradley
Advanced Processor Design
Entry Systems Division, IBM (Boca Raton, Florida)

Prevod: *Aleksandar Kragl*

BRZE MREŽE

Poslednja dekada ovog veka protiče u znaku telekomunikacija i računarskih mreža. Ubrzano se radi na implemetaciji mreža valikih brzins, takozvanih gigabit mreža koje, zajedno sa napretkom ostvarenim u multimediji, polako brišu granicu izmedu telekomunikacija i računarskih komunikacija.

Aleksandar Radovanović

Napredak ostvaren u razvoju hardvera računarskim komunikacijama i telekomunikacijama uticao je na istraživanje i razvoj gigabit mreža. Primetan je ogroman napredak u računarskoj moći i širini propusnog opsega komponenata računarskog sistema.

Performanse procesora gotovo da se udvostručavaju svake godine. U korak ide i napredak u oblasti paralelnog procesiranja. Penlenje su takođe sve bojih performansi. Računarski sistem se već danas sastoje od većeg broja procesora sposobnih za obradu podataka reda gigabita u sekundi (Gbps), penlenja kao što su nizovi diskova (*disk arrays*) i grafičkih sistema sposobnih da prime ogromne količine podataka.

Gledajući iz softverske perspektive, sve veća računarska moć omogućava pojavu niza aplikacionih programa. Računari današnjice dovoljno su moćni da podržavaju niz interaktivnih programa kao što su simulacije, multimedija, video konferencije ili virtualna stvarnost.

Porast moći hardvera i softvera utiče i na razvoj računarskih mreža koje moraju biti u stanju da podrže čitav niz novih zahteva. Jedna od najuspešnijih ideja je svakako koncept povezivanja lokalnih računarskih mreža u veće celine (*Internetworking*). Telekomunikacione mreže, telefonska na primer, sve su se više koristile za prenos podataka. Rezultat je nestajanje granice između računarskih i telekomunikacionih mreža. Interesi tri grupe: računarske, računarskih telekomunikacija i telekomunikacija polako su se stopele u zajednički. Sve tri danas su zainteresovane za podršku novom tipu usluga koji kombinuje računare, računarske mreže i korisnike. Kao rezultat nastaju gigabit mreže.

GIGABIT LAN, MAN i WAN

Davaoci telekomunikacionih usluga dugo su smatrali da brzina prenosa od 56 ili 64 kilobita u sekundi (Kbps) zadovoljava prosečnog korisnika. To je i bio slučaj za mainframe računarske sisteme bazirane na terminalima i *time-sharing*-u. Sa rastućim brojem personalnih računara i pojavom LAN-a (*Local Area Network*) dolazi do razvoja takozvanog klijent server modela. Trenutno zahtevi koje ovaj model postavlje zadovoljeni su LAN-om. U klijent server sistemima većina zadataka se obavlja lokalno na stonom računaru. Od udaljenog servera se obično traži prenos veće količine podataka, postavlje se zahtev za proračune većeg obima, kvalitetnim štampanjem i slično. U toku rada klijent ne mora da zna kada i od kog servera će zahtevati uslugu. Uobičajeno je da su klijenti radne stanice na kojima rade *multitasking* operativni sistemi tako da različiti programi mogu u isto vreme zahtevati usluge od različitih servera, pa čak i procesi u istom programu mogu da se obraćaju nekolicini servera. Očigledno je da je većina današnjih telekomunikacionih mreža potpuno neupotrebljive u ovom modelu.

Lokalne se mreže povezuju u veće celine MAN i WAN (*Metropolitan Area Network* i *Wide Area Network*) preko kojih razmenjuju podatke koristeći postojeće sisteme slabih performansi. U isto vreme viši se pritisak na telekomunikacione kompanije da poboljšaju svoje usluge u smislu uvođenja većih brzina i pokrivanja šireg prostora optičkim kablovima. Optički sistemi trenutno omogućavaju prenos podataka brzinama od 2.488 Gbps na daljinama do 80 km ili većim uz upotrebu *repeater*-a. Optoelektronski elementi za sada zadovoljavaju zahteve za prenosom od 10 Gbps na razdaljinama od 35 do 40 km.

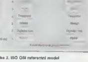
GIGANIT APLIKACIJE

Gigabit mreže otvaraju put razvoju i primeni novih aplikacija koje se grubo mogu podeliti u dve grupe. Prva se odnosi na distribuirane sisteme koji zahtevaju dobre performanse između računara. Drugu grupu čine interaktivne distribuirane aplikacije sa korisnikom koji traži informacije brzinom kojom može da ih obradi.

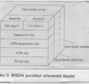
Sigurno najatraktivnija gigabit aplikacija je takozvani *video-on-demand*. Filmovi se nalaze na optičkim diskovima na jednom mestu, a korisnici sa svojih radnih stanica sistemom menija biraju film koji žele da pogledaju. Televizija visoke definicije – HDTV (*High Definition TV*) zahteva prenos podataka brzinom većom od 1 Gbps. Kompresijom se ovaj zahtev svodi na nekoliko Mbps. Ukoliko je na mrezi u isto vreme više korisnika koji zahtevaju različite filmova brzina prenosa se opet penje na gigabite u sekundi. Drugi važan zahtev koji se postavlja pred mrežu je minimalno kašnjenje. Ukoliko slike zakasni i deo sekunde korisnik to primecuje.

Ne manje atraktivne aplikacije je video-konferencije. Radne stanice korisnika opremljene su kamerama, zvučnicima i mikrofonima zahvaljujući kojima stupaju u kontakt sa ostalim korisnicima mreže. Savremene radne stanice, na primer Sun ili HP, u standardnu opremu uključuju multimediju, a u poslednje vreme sve je više PC računara sa sličnim hardverom.

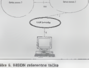
Simulacije i generisanje slike tipične su gigabit aplikacije. Na primer, distribuirani meteorološki sistem prikuplja podatke na različitim udaljenim lokacijama. Centralni kompjuter u realnom vremenu uzima podatke sa radnih stanica i generiše sliku. Prvi deo procesa, prikupljanje podataka, primer je distribuiranog sistema velike brzine. Isto tako, distribucija slike zahteva mrežu velike brzine.



Slika 2. ISO OSI referentni model



Slika 3. BISDN protokol referentni model



Slika 5. BISDN referentne tačke

OSI ARHITEKTURA

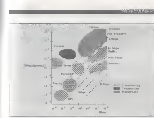
Arhitektura mreže definiše funkcije koje ona i njene komponente treba da izvršavaju i okvir u kome će računari različitog hardvera i softvera uzmati učešća u mreži. Da bi se pojednostavio dizajn i složenost, većina mreža je organizovana kao niz slojeva. Svrha svakoga od njih je da ponudi izvesne usluge-servis sloju iznad sebe ne opterećujući ga pri tome detaljima kako je taj servis implementiran. Dva sloja implementirana da dva različita računara na različitom hardveru i softveru komuniciraju pridržavajući se niza pravila koja se nazivaju protokol. Protokoli zajedno sa slojevima nazivaju se mrežnom arhitekturom. Danas se u svetu koristi nekoliko arhitektura kao što su SNA, TCP/IP ili OSI.

Još je 1984. godine ISO (*International Organization for Standardization*) objavila standard nazvan OSI (*Open Systems Interconnection reference model*). Referentni model se sastoji od sedam slojeva:

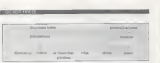
7. Aplikacioni sloj pruža usluge kao što su elektronska pošta, prenos datoteka, pristup udaljenom računaru itd.
6. Prezentacioni sloj konvertuje podatke primljene u standardnom formatu u format koji odgovara računaru na kome je sloj implementiran. Npr. IBM sistem konvertuje tekst iz EBCDIC koda u standard za OSI, podaci se zatim prenose mrežom do PC-ja koji ih iz OSI standarda pretvara u ASCII i prosleđuje aplikacionom sloju da ih interpretira i prikaže na ekranu.
5. Sloj sesije uspostavlja, održava i prekida vezu dva aplikaciona programa.
4. Transportni sloj omogućava pouzdani prenos podataka, kontrolu njihovog toka i oporavak od grešaka.
3. Mrežni sloj je obično implementiran van korisnikovog računara. Sa jedne strane ovog sloja je korisnik a sa druge mreža. Ovaj je sloj odgovoran za uspostavljanje, održavanje i prekidanje veze na samoj mreži. Slojevi iznad ovoga ne vide mrežu, vide samo odgovarajući sloj na nekom drugom računaru poznate adrese.
2. Sloj pouzdane digitalne veze zadužen je za pouzdan prenos podataka preko fizičke veze. On šalje blok podataka i vodi računa o sinhronizaciji, kontroli grešaka i kontroli toka. Često se smatra da je transportni sloj nepotreban ukoliko ovaj sloj dobro vrši svoju funkciju.
1. Fizički sloj je zadužen za prenos niza bitova preko fizičke veze. Sloj operiše sa bitovima kao sa signalima određenog napona i trajenja. Za ovaj sloj su definisani standardi koji opisuju električne, mehaničke i proceduralne karakteristike veze. Na primer nivoe napona za bitove, oblik i broj nožica na konektorima itd.

INTEGRISANI SERVISI

CCITT (*International Consultative Committee on Telegraphy and Telephony*) 1984. godine predstavlja niz standarda namenjenih prenosu glasa i podataka.



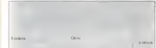
Slika 1 Brzina odziva različitih sistema



Slika 4. Komutacione tehnike



Slika 7 Virtuelni kanal



Slika 10 Logička struktura SONET okvira



Slika 9 Virtuelni put

Želje CCITT-a bila je da definiše ISDN (*Integrated Services Digital Network*) mrežu koja će definitivno izbrisati granicu računarskih komunikacija i telekomunikacija i pružiti korisniku niz novih usluga. Pokazalo se da je želja bila preuranjena jer ISDN mreža nije zaživela. Pojavljuje se LAN (*Loacal Area Network*) i sve mu se čini da je predlog CCITT-a o prenosu podataka brzinom od samo 64 Kbps u vreme kada se opanselo brzinama od desetine megabajta u sekundi (Mbps) zastareo. Međutim CCITT pravi još jedan pokušaj 1989. i 1990. godine kada donosi novi *Broadband ISDN* (BISDN) standard. BISDN za razliku od svojih prethodnika ide u korak sa vremenom i definiše prenos podataka brzinama od nekoliko stotina Mbps. Prenos informacija se zasniva na ATM (*Asynchronous Transfer Mode*) i OSI arhitekturi. Referentni BISDN model razlikuje tri ravni:

1. Ravan korisnika koja se bavi prenosom informacija.
2. Kontrolna ravan je odgovorna za uspostavljanje i kontrolu veze.
3. Upravljačka ravan uključuje upravljanje sa druge dve ravni i koordinaciju među slojevima.

a) ATM adaptacioni sloj se koristi za transfer informacija koje nisu bazirane na ATM-u, npr. video signal.
b) ATM sloj vrši neposredan prenos informacije.
c) Fizički sloj se bavi prenosom signala.

KOMUTACIONE TEHNIKE

Danas se primenjuje nekoliko tehnika kojima se informacije prenose kroz telekomunikacione mreže. Dve ekstremne sa slike predstavljaju dve tradicionalne tehnike: komutaciju vodova i komutaciju paketa. Suština tehnike komutacije vodova (*circuit switching*) je da se pre početka razmene podataka uspostavlja stalna veza fiksnog kapaciteta. Kod komutcije paketa podaci se rastavljaju u takozvane pakete koji nezavisno putuju od noda do noda u mreži i dešava se da na odredište ne stignu redom kojim su poslati. Komutacija ćelija (*cell relay*) ili kako se često naziva ATM (*Asynchronous Transfer Mode*) predstavlja kulminaciju razvoja komutacionih tehnika poslednjih dvadeset godina.

ATM je tehnika komutacije dizajnirane za prenos kratkih paketa podataka preko Gbps veze velikih dužina. ATM mreže slično današnjim telefonskim mrežama hijerarhijski su organizovane. Korisnik se uključuje na lokalnu mrežu na svom kućnom priključku, a lokalna se mreža povezuje na nacionalu i internacionalnu. Korisnikov interfejs prema mreži naziva se UNI (*User Network Interface*) (T tačka na slici 5). Tačka S sa slike predstavlja granicu između lokalne mreže i korisnikove multimedia stanice. Veza između regionalnih i nacionalnih mreža ostvaruje se preko NNI-a (*Network Network Interface*).



Slika 6 Format ATM ćelije



Slika 8 Virtuelni put

ATM VEZA I VELIKE BRZINE

Ogromna brzina prenosa podataka postiže se zahvaljujući malim paketima podataka, ATM ćelijama. Kod ove vrste prenosa podrazumeva se korišćenje optičkih kablova velika pouzdanosti. Ovim je dobijeno na brzini procesiranja ćelija jer je provera grešaka svedena na minimum. Slika 6 (a) prikazuje format ATM ćelije. U CCITT terminologiji bajtovi se nazivaju oktetima pa su tako i obeleženi na slici. ATM ćelija se sastoji iz dva dela: zaglavlja i informacionog polja. Sa slika 6 (b) i 6 (c) se vidi da postoje dve vrste zaglavlja: UNI i NNI zaglavlje. Funkcija pojedinih polja biće jasnije ako posvetimo nekoliko reči o načinu uspostavljanja i održavanja veze.

Pre negošto se pošalje ijedna ćelija veza mora biti uspostavljena između predajnika i prijemnika, što nije bio slučaj pri paketnom prenosu. ATM sloj (slika 3) na obe strane je zadužen za održavanje uspostavljene veze. Sam ATM sloj sastoji se iz dva podsloja: virtuelnog kanala (VC – *Virtual Channel*) i virtuelnog puta (VP – *Virtual Path*). Ovi su slojevi namenjeni održavanju dve vrste veze: VCC (*Virtual Channel Connection*) i VPP (*Virtual Path Connection*). VCC je jednostavno veza između dve korisnika u mreži koji razmenjuju ATM ćelije u oba smera. Svaki od njih ima svoj identifikator obeležen kao VCI (*Virtual Channel Identifier*). Identifikator nije jedinstven za vezu već se menja pri svakom prolazu kroz komutacionu tačku. VC je u stvari opisan nizom VCI-a. Slika 7 prikazuje primer funkcionisanja jedne ovakve veze. Računar A je zahtevao vezu sa B. Mreže mu dodeljuje VCI a koji će on koristiti za upis i čitanje podataka. Mrežni softver nalazi put do računara B i pri prolazu kroz komutacione čvorove (neke druge računare) menja ovaj VCI u neki drugi. Tako a na čvoru L1 postaje b, na čvoru L2 x i na L3 y što se biti i VCI za računar B. Svaki od čvorova pravi ulazno-izlaznu tabelu koja se koristi tokom trajanja veze.

Skup VCC-a sa istim krajnjim tačkama grupiše se u VPC (slika 8). Kao i za VCC, tako se i za VPC

Nastavak na strani 124

Nastavak sa strane 45

Pristup disku sa uključenim FastDiskom, bez VFAT-a

Svi Int 13h pozivi se obrađuju u FastDisk veznicima. U standardnom slučaju sa diskom komunicira WDCTRL veznik, naravno ako je kontroler diska kompatibilan sa WD 1003 standardom. U sekciji [386Enh] nalaze se sledeće linije

```
device =ifsmgr.386
device=*blockdev
32bitdiskaccess =on
device=*int13
device=*wdctrl
```

U ovom slučaju se obrada Int 21h poziva obrađu je u MS-DOS-u i sa veznicima za realni način rada.

Pristup sa uključenim FastDiskom na disku koji je podignut preko VFAT-a

Ovo je najbolja varijante za standardan disk kontroler. Uključen je *32-bit Disk Access* i *32-bit File Access*, pa se svi pozivi Int 13h obrađuju u 32-bitnom WDCTRL vezniku, a svi pozivi Int 21h u vezniku VFAT, takođe u zaštićenom načinu rada. U sekciji [386Enh] nalaze se sledeće linije

```
device=ifsmgr.386
device=*blockdev
device=*int13
32bitdiskaccess=on
device=*wdctrl
device=vfat.386
device=vcache.386
```

Pristup sa VFAT-om, isključen FastDisk

Ovo je slučaj kada se *32 bit Disk Access* ne može uključiti – na primer na računaru sa SCSI diskom i adapterom. U ovom slučaju se instalira modul za mapiranje poziva (RMM D32). Pozive Int 21h obrađuje VFAT, a onda se propuštaju kroz modul za mapiranje do MS DOS veznika. U SYSTEM.INI datoteci ce sekcije [386Enh] sadržavati sledeće linije

```
device=ifsmgr.386
device=ios.386
device=vxdldr.386
device=vfat.386
device=vcache.386
```

Komprimovani disk podignut pomoću VFAT-a, bez FastDiska

Ponovo se instalira RMM D32, a pozive Int 21h obrađuje VFAT pre nego što se pošalju kroz modul za mapiranje prema MS DOS u njegovim veznicima. Nakon obrade u MS-DOS-u, podaci se propuštaju kroz veznike softvera za kompresiju. U SYSTEM.INI datoteci ce sekcija [386Enh] sadržavati sledeće linije

```
device=ifsmgr.386
device=ios.386
device=vxdldr.386
device=vfat.386
device=vcache.386
```

Komprimovani disk podignut pomoću VFAT-a, uključen FastDisk

Ovo je najbolja varijanta kada je disk komprimovan. Instaliran je modul za mapiranje (RMM D32). VFAT obrađuje Int 21h. Ako se poziv odnosi na komprimovan disk, zahtev se propušta kroz RMM D32 do veznika za kompresiju. Nakon što MS DOS veznici završe sa podacima, informacije se vraćaju na 32 bitni veznik za *FastDisk* pristup disku, koji se direktno obraća kontroleru. U SYSTEM.INI datoteci će sekcija [386Enh] sadržavati sledeće linije

```
device=ifsmgr.386
32bitdiskaccess=on
device=*int13
device=*wdctrl
device=ios.386
device=vxdldr.386
device=vfat.386
device=vcache.386
```

Sve u svemu, arhitektura *Windows for Workgroups 3.11* svakako je napredak u odnosu na verziju 3.10. Držim da ovaj paket treba shvatiti kao najavu mogućnosti koje će pružiti *Chicago*, eventualno najavljeni paket *Daytona* ili *Windows NT 3.11*. Ono što ćete primetiti već pri instalaciji je mogućnost izostavljanja mrežnih sposobnosti i konfigurisanje za upotrebu na individualnom računaru. Prema tome, ovaj *Windows* je prilika da unapredite svoju kućnu mašinu i čekate na pojavu *Chicaga*.

STO BROJEVA U 16 SLIKA

Krajem decembre 1983. godine na kioscima širom zemlje se pojavio prvi broj časopisa „Računari" koji je, pod nazivom „Računari u vašoj kući", izašeo kao specijalno izdanje časopisa „Galaksija". Izdanje je osmislio i uredio Jovan Regasek, sve tekstove u njemu napisao Dejan Ristanović, a lansiran je i projekat malog kućnog računara „galeksija", koji je konstruisao Voja Antonić. Tako je počelo, a ostalo je... istorija. Listajući malo po svojoj memoriji, a malo po starim godištima česopisa, Dejen Ristanović, autorska kičma lista od prvog do stotog broja, vodi vas ovim tekstom na melo putovanje kroz vreme. Dragi čitaoci, vežite pojaseve. Polećemo!

Nije lako poverovati da je za „Računarima" više od deset godina i tačno 100 brojeva – a kao da smo juče po prvi put listali „Računare u vašoj kući" i posmatrali prvo READY koje je ispisao računar „galaksija"! Prelistao sam devet ukoričenih godišta „Računara", podsetio se mnogih dragih tekstova, nasmejao se raznim prepucima i pitao se kako sto brojeva sažeti na par sledećih stranica. Najzad sam se setio Džordža Lukasa koji kažu, scenario za film ne piše nego ga crta – svaki film se svede na nekoliko slika, a ono „između" se posle lako rekonstruiše. I zaista, u razvoju jednog časopisa zapažaju se pojedini prelomni trenuci, koncepcijske raskrsnice sa kojih se moglo krenuti raznim putevima i tako stići na razna mesta. Možda je neki vešt „istončar budućnosti" još prvih dana 1984. godine mogao nacrtati scenario naših sto brojeva – u 16 slika!

Dejan Ristanović

SLIKA 0000: Prolog

Očekivalo bi se da priča o nekom časopisu počne od njegovog prvog broja, ili barem od priprema za njega; priča o „Računarima" počinje hm ili četiri godine ranije, na stranicama časopisa „Galaksija". „Galaksija" se, kao „Računari" sada, bližila stotom broju. Osmogodišnji predan rad na popularizaciji nauke, pod „dirigentskom palicom" osnivača i dugogodišnjeg glavnog urednika **Gavrila – Gaje Vučkovića**, tih dana je dobijao najviše ocene domaće naučne i stručne javnosti. Početkom 1980. godine „Galaksija" je, uz astronomiju i astronautiku koje su od samoga početka preovlađivale, iz okosnicu svakog broja, počela da posvećuje sve više pažnje i praktičnim temama. U okviru tog bloka, koji je uređivao **Jovan Regasek**, inače tada urednik za prirodne nauke, februara 1980. godine pokrenuta je rubrika „Nauka i igra" koju je vodio dr **Vladimir Ajdačić**. Rubrika se pretežno bavila matematikom „skrivenom" iza nekih popularnih igara; naročitu pažnju privukao je šah i specijalizovani šahovski kompjuteri koji su u to vreme sticali popularnost – neki od takvih računara „ukrstili su koplja" pred čitaocima, što je izazvalo veliko interesovanje i neminovna pitanja tipa „kako mašina može da 'misli'?" sjajne primamina tematici koja je interesovala „Galaksiju". To je ujedno bila i premijera računarstva na stranicama „Galaksije" i, što se toga tiče, čitave domaće popularne publicistike.

Što se mene tiče (ovo je, znate, dobrim delom lični pogled na proteklih petnaestak godina i ne predstavlja zvaničnu istoriju časopisa), u to vreme bio sam učenik četvrtog razreda Matematičke gimnazije i uživao sam da rešavam nagradne matematičko-logičke probleme u časopisu „Galaksija". Usudio sam se da jedno od pisama završim pominjanjem džepnih računara sa mogućnošću programiranja na kojima takođe postoje igre, možda malo jednostavnije od šaha ali itekako zanimljive. To je bilo sasvim dovoljno profesoru **Ajdačiću** da me pozove da napišem tekst o džepnim programabilnim računarima koji će, pod naslovom „Igre pomoću računara", biti objavljen januara 1981. godine u „Galaksiji 105".

Čitanje tog teksta trinaest godina kasnije bilo je svakako zanimljivo iskustvo. Iz njega sam, na primer, saznao da sam se kao dete igrao klikera, što je, garantujem, čista dezinformacija: verovatno se neko iz Redakcije potrudio da mom previše tehnički orijentisanom tekstu napravi poetski uvod. Ali, uz taj uvod, predstavljanje tadašnjih „zvezda" računara *Texas Instruments 59* i *Hewlett Packard 41C* i opis igre „Most" (trodimenzionalni *Tic-tac-toe*), u tekstu se javlja jedna dobra ideja: osnivanje Kluba koji će okupiti vlasnike džepnih programabilnih računara. Iako su ti računari bili relativno skupe i teško nabavljive „igračke", pokazalo se da ih po Jugoslaviji nije malo – počela su da stižu pisma sa originalnim prilozima naših čitalaca, pa je već u „Galaksiji 112" formirana svojevrsna biblioteka programa koja je funkcionisala na principu razmene. Katalog je iz meseca u mesec rastao i krajem 1983. godine prikupio preko 1000 naslova; neki od domaćih programa bili su zaista svetski napisani, i kasnije preneseni u ugledne strane časopise slične tematike.

Pa, ipak, zvezda džepnih programabilnih kalkulatora polako je tamnela. Zasenice ih jedva nešto veći računar koji je **Stanko Popović** predstavio jula 1982. godine u „Galaksiji 123". Tekst se završavao rečenicom „Spectrum je računar pred kojim je sjajna budućnost". Budućnost koja je upravo počinjala!

SLIKA 0001: "Računari u vašoj kući" (januar 1984)

Ideja o specijalnom izdanju časopisa „Galaksija" koje bi bilo posvećeno isključivo kompjuterima rodila se u leto 1983. godine. Interesovanje čitalaca bilo je veliko, tržište je prosto vapilo za domaćom literaturom, a i u samoj Redakciji se pre svega kroz blok „Igre" hobi urad sam ... kao ... dva specijalna izdanja druge tematike, stekło dosta iskustva za jedan takav projekat. Urednik tog bloka, a zatim i specijalnog izdanja bio je **Jovan Regasek**, koji će, u svojstvu glavnog i odgovornog urednika, voditi „Računare" kroz sva sela i hanbe ovih deset godina i uspešno doploviti do stotog broja.

Prvobitno zamišljeni „Računari u vašoj kući" prilično se razlikuju od izdanja koje se krajem 1983. godine pojavilo na kioscima. Planirano je da polovina specijalnog izdanja bude posvećena džepnim, a polovina kućnim računarima, ali se tokom pripreme iskristalisalo mišljenje da se treba okrenuti budućnosti i onome o čemu ranije uopšte nismo pisali: deo posvećen džepnim računarima, premda napisan, ostao je „u fioci". Tako je oblikovano izdanje u kome će svako naći po nešto: oni koji tek planiraju nabavku kompjutera testove (tada) popularnih modela, oni koji su tek nabavili kompjuter školu bejzika, bolji poznavaoci „mala tajna velikih majstora programiranja", a oni koji su spremni da uzmu lemilicu u ruke – a za njih smo imali pravu poslasticu u vidu prvog domaćeg kućnog računara!

Računar „galaksija" nastao je sredinom 1983. godine. Njegov konstruktor **Voja Antonić** u ovom projektu je sjedinio dobre osobine prethodnih nekoliko kompjutera koje je konstruisao, trudeći se da računar što više pojednostavi i učini ga pogodnim za samogradnju. Naše specijalno izdanje je bila prilika da se čitaocima, uz obilje praktičnih saveta, pruži mogućnost da potpuno legalno dođu do svog prvog kompjutera. Jer, uvoz svega vrednijeg od 5000 dinara (u našim parama oko 140 maraka) tada je bio strogo zabranjen, a uvozne komponente za „galaksiju" mogle su se spakovati tako da koštaju manje od toga. Tokom poslednjih meseci 1983. godine **Voja Antonić** će dalje usavršiti hardver i (posebno) softver računara „galaksija" da bismo u „Računarima u vašoj kući" pred čitaoce izašli sa jednim zagrednju jednostavnim ali ipak (za to vreme) dovoljno moćnim kućnim kompjuterom.

„Računari u vašoj kući" su naprosto planuli – samo mesec dana kasnije morao je da se doštampava još jedan tiraž, a kasnije je izdanje još jednom ponovljeno. Ostali mediji su se zainteresovali za kompjutere, snimljene su brojne televizijske i radio-emisije (posebno mesto svakako zauzima „Ventilator" **Zorana Modlija** koji je, valjda po prvi put u istoriji radija, na ideju urednika „Računara" emitovao kompjuterske programe preko radio-talasa) pa su čak i vlasti pokazale malo sluha i (delimično) izuzele kompjutere od strogih uvoznih zakona. Računarsku revoluciju na ovim prostorima više ništa nije moglo da zaustavi.

Najprijatnije iznenađenje je svakako predstavljao računar „galaksija": više od 8.000 čitalaca naručilo je komplet delova i tokom sledećih meseci uspešno sastavilo svoj prvi kompjuter. Taj broj bio je toliko veći od svega o čemu smo uopšte mogli da pomišljamo da se naša mala Redakcija našla u svojevrsnom „opsadnom stanju": pisma, narudžbenice, doznake, EPROM-i, saveti, telefoni mesecima nisu prestajali da zvone. Kroz sve te meseca počeli smo da shvatamo jednu važnu stvar: izdavanjem „Računara u vašoj kući" nismo samo završili jedan veliki posao – naprotiv, počeli smo jedan mnogo veći!

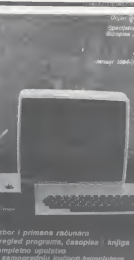
SLIKA 0010: Jednom mesečno („Računari 8", oktobar 1985)

Stara je istina da su prvi brojevi nekog časopisa presudni za njegovu koncepciju, pa je možda druga slika ovoga scenarija trebala da se bavi „Računarima 2". „Računari u vašoj kući", naime, i nisu bili časopis u pravom smislu te reči: smanjite im format, napravite čvršće korice, izbacite par kupona koji se zaseckuju i ... eto knjige. „Računari 2" izašli šest meseci kasnije, već su imali obeležja časopisa – napisalo ih je više saradnika, tematska lepeza tekstova bila je šira, opisani su neki aktuelni sajmovi u zemlji i svetu, pojavili su se i prvi oglasi ... pa ipak, „Računari" su u svojoj prvoj godini izlazili sasvim neredovno, što teško da je osobina jednog „pravog" časopisa. Počevši od „Računara 8" izlazimo jednom mesečno, pa je možda taj trenutak vredan posebnog pomena.

U prvih osam brojeva „Računara" desilo se puno zanimljivih stvari – samogradnja „galaksije" uspešno je okončana, konstruisana je „Galaksija Plus", detaljno su testirani svi tada popularni kućni računari, proučen ROM svakog od njih, udaren temelj budućih serija tekstova o fundamentalnim tehnikama programiranja, najavljen prvi program koji će biti izdat na kaseti: „Ekranski editor" **Vlade Kostića**. „Računari" su, kratko rečeno, već uveliko poprimili svoj kasniji oblik, a doživeli smo i prvu pravu polemiku!

Povod za polemiku bio je razgovor sa profesorom **Jozom Dujmovićem**, objavljen u „Računarima 5". Po prvi put su na stranicama našeg časopisa čitaoci imali priliku da pročitaju nešto što možda **nisu** želeli da čuju: umesto glorifikacije bejzika i „mašinca" kao i programera koji, pošto su sami naučili da programiraju na ovim jezicima, sanjaju da će baš sutra prodati svoju novu akcionu igru Englezima, profesor **Dujmović** je izrazio uverenje da su ti „samouki hakeri" po svemu sudeći izgubljena generacija, a da softverska šansa leži u projektovanju ozbiljnih aplikacija, operativnih sistema, kompajlera i uopšte poslu koji mogu da obave samo visoko školovani inženjeri računarske tehnike. Kao svojevrsnu „kontratežu" ovom intervjuu „Računari" su objavili tekst „Pravi programeri ne govore paskal", neku vrstu manifesta u kome se, uz dosta esnafskih šala, tvrdi da su jedini jezici vredni truda fortran i asembler (bejzik, eto, ni 'tu nije pomenut), a da su paskal i slični strukturirani jezici dobri tek za knjiške moljce ili, kako je tamo rečeno, „zderače pita".

I o intervjuu i o manifestu se mnogo pričalo, kako po programerskim kuloarima tako i na stranicama „Računara" – bio je to „putokaz" Redakciji da su intervjui štivo koje čitaoce veoma zanima. Sledeći intervju bio je prilika ne samo da se čuju zanimljive tvrdnje, nego i da „Računari" dobiju jednog izuzetnog saradnika – profesor **Dušan Slavić** pristao je da sa našim čitaocima podeli rezultate svog dugogodišnjeg rada na numeričkim algoritmima pa smo kroz nekoliko serija tekstova saznali kako se na kompjuteru računaju



vrednosti elementarnih funkcija uvideli da je matematički segment sistemskog softvera većine kućnih računara traljavo urađen upoznaš najvažnije numeričke algoritme lei su to tekstovi na najvišem nivou kojima su se „Računari" ikada ponosili. Na žalost, tragična smrt profesora **Slavića** prekinula je ovu seriju na najzanimljivijem mestu.

Informativni segment „Računara" bio je od samog početka veoma jak pre svega zahvaljujući prilozima iz Londona koje je slao **Anđelko Zgorelec**, čovek koji je nekoliko godina ranije obnevao čuveni časopis *Personal Computer World* i koji se nalazio u samom centru evropskih kompjuterskih događaja. Prilozi gospodina **Zgoreleca** omogućili su čitaocima „Računara" da zavire iza kulisa kompjuterske industrije i saznaju kakve sile pokreću stvari i šta se može očekivati u neposrednoj budućnosti „uzgred" su saznavali i sve o onome što se dešavalo u sadašnjosti.

SLIKA 0011: Noge („Računari 11", januar 1986)

„Računari" se od prvih dana odlikuju naslovnim strana ma koje retko koga ostavljaju ravnodušnim. Čuli smo o nji ma svašta: da su dobre, da privlače poglede (a ponekad i kupce), da su grozne, da ne priliča jednom ozbiljnom časopisu, da su dobra ideja, da bi trebale da budu „slobodnije" ... priči nikad kraja! A retko će se koji časopis buniti što se o njemu piše stara istina je da je i kritika bolja od ravnodušnosti pa su se i naslovnim stranama postale golovo zaštitni znak „Računara".

Od svih devedeset devet naslovnih strana najčuvenija je ona sa „Računara 11" zvana kratko „Noge" i noga baš i nisu dosežu otprilike do Amstrada koji se namestio ispred njih) kompliment koji su im čitaoci uputili. Zamerene su kvrgave previše mišićave a sandale i „haljinu" da i ne komentarišemo.

Nije malo onih koji su nam poleskali pomislivši da smo lom naslovnom stranom hteli da „uzburkamo vodu" tj. da malim skandalom privučemo pažnju publike. Ništa od toga naslovna strana je prosto loše ispala. Na slajdu je izgleda la relativno korektno ali kada je došla na papir katastrofa. Zapravo još i veća katastrofa nego što čitaoci znaju jer je tek krajnjim naporima montaža uspela da „povuče" sliku tako da se pri obsecanju ukloni njen vrh na kome se lepo video i izvestan deo garderobe koji se obično nosi ispod haljine. No iz ove kože se ne može pa se trebalo prilagoditi situaciji što je izvanredno obavio **Branko Đaković** kroz nekoliko brojeva je „podizao temperaturu" oko tih nesrećnih Nogu da bi najzad raspisao konkurs „Čije su Noge sa naslovne strane". Velika većina čitalaca je iz samo njima znanih razloga bila ubeđena da Noge sa naslovne strane pripadaju **Jeleni Rupnik**, našoj saradnici koja je pisala kontraverzne tekstove počevši od čuvenog „Računar je muškog roda". Bilo kako bilo o Nogama se pričalo i pisalo. I ljudi ih se dan danas osam godina kasnije, sećaju i možda očekuju od mene da u ovom jubilarnom broju otkrijem njihovu poslednju (?) tajnu. Pa neka bude: nisu to noge **Jelene Rupnik** pa čak ni neke od tadašnjih (ili dalekosile sadašnjih) estradnih zvezda. Vlasnica Nogu je „samo" supruga fotografa koji je usilkao naslovnu stranu.

Kasnije će se pokazati da su Noge i po drugim osnovama važan trenutak u istoriji „Računara" označile su snažnije okretanje čitaocima. Videvši da su čitaoci nekako raspoloženi da nam pišu trudili smo se da ih u tome stimulišemo, pa se pojavila rubrika pisama, Peek & Poke Show, razni spiskovi „poloca", Fitalice, P. A. Marvin, crteži čitalaca i svašta drugo. To su bili dani u kojima ste se listajući „Računara" mogli zaista slatko nasmejati.

To dakako ne znači da u „Računarima" nije bilo i ozbiljnih stvari – zapravo baš je od „Računara 11" da ne kažem od Nogu počela zapažena serija tekstova o teoriji programiranja i algoritmima vremenom smo detaljno „pokrili" dobar deo stvari o kojima su Knuth, Dijkstra i drugi pisali u svojim slavnim knjigama naravno na nivou koji odgovara našim čitaocima. Nekoliko sledećih brojeva donece „Radionicu logičkih igara", nastavak serije tekstova o numeričkim algoritmima i sistemskom programiranju. Kada se radi o popularnim kompjuterima Spectrum i Commodore 64 bili su na vrhuncu slave. Zapažene tekstove o Komodoru 64 pisao je **Zoran Životić** dok su se Spektrumom najozbiljnije bavili **Vlada Kostić** i **Jovan Skuljan**.

SLIKA 0100: Neka bude IBM („Računari 14", april 1986)

Spectrum je rekoh na vrhuncu slave, šta znači da je pravi trenutak da se pojavi nešto novo. I to novo se zaista pojavljuje predstavljeno možda najznačajnijim tekstom koji su „Računari" ikada objavili tekstom koji će odrediti čitavu budućnost časopisa, tekstom „Neka bude IBM" (naslov dala redakcija) **Srđana Koseca**, koji je objavljen u „Računarima 14". Mala sve češća čitalaca su tim tekstom dobile spornu temu za razmišljanje: dolazi li to trenutak da se kućni računar za nešto i primeni?

Daleko lalo da smo mi govorili da se i tadašnji Spektrumi i Komodori ne mogu korisno upotrebiti – zapravo „Računari" su vrveli od njihovih korisnih primena ali ovako ruku na srce sva ta korišćenja su bile dosta slabe. I to ne samo krivicom tadašnjih kompjutera: za pravu primenu osim računara je potrebno još dosta toga: periferije, diskete, diskovi, štampači, modemi. IBM PC kompatibilni računari znečili su većinu potrebnih i međusobno kompatibilnih dodataka u jednoj kutiji uz odgovarajuću softversku podršku. I to po ceni koja se može podneti.

Naglasak je na reči *kompatibilnih* – pravi IBM PC računari bili su mnogo skupi, pa su se Jugosloveni uglavnom opredeljivali za tajvanske kopije. U to vreme kloneri su bili prava „mačka u džaku" koju je trebalo sklapati od delova kupljenih na raznim mestima i to uz stalne brige šta će se dogoditi na našoj carini. Zato su relevantne informacije sa strane „Računara" imale ogroman značaj – svojoj srećnoj zvezdi (i znanju gospodina **Koseca**) imamo da zahvalimo što je baš prvi objavljeni tekst bio maksimalno kompetentan. Slobodno ga i danas pročitajte: nećete naći ni reč koja ne bi bila tačna ili koja i dan-danas ne bi važila, ali je još fascinantnije što su i prognoze bile dobre. Ko zna koliko bi se na primer kupaca „nasankalo" na CGA karticu da u tekstu nisu jasno objašnjene prednosti „nekompatibilnog" Herculesa.

Tekst je, rekoh, stavio ljudima „bubu u glavu" i u pravcu Minhena su krenule prave pravcate „PC ekspedicije" ko se od tamošnjih prodavaca snašao da obezbedi nekog službenika koji govori srpski i pristojno osiromašuje jugoslovenskim džepovima napravio je dobar posao. U zemlju je iz dana u dan stizalo sve više i više PC računara što se, naravno, itekako odrazilo i na koncepciju našeg časopisa.

Spectrum i Commodore „vladali" su i u „Računarima 14" i u „Računarima 16" ali je bilo sve očiglednije da im se „mandat" bliži kraju. Nekim čitaocima nije bilo lako da se sa tim pomire – rubrika Load „Drugi Računari" bila je puna gnevnih pisama u kojima se „normalnost" Redakcije stavlja pod ozbiljnu sumnju: ko je lud da piše o nekim basnoslovno skupim računarima koje „niko živ" nema pored toliko hiljada Spectruma i ljudi koji se za njih interesuju. Samo snobovi poput nas rakoša nam gnevni uglavnom mlađi čitaoci. Budućnost je, ako nam oprostite na neskromnosti, pokazala koliko smo bili u pravu – sada više nije važno da li smo taj zaokret napravili zato što smo anticipirali pravi bum personalnih računara u Jugoslaviji, da li zato što smo bili svesni svoje snage da taj bum (delimično) i sami izazovemo ili, možda zato što smo o Spectrumu i drugim kućnim računarima napisali (gotovo) sve što se moglo napisati. Važno je

su se „Računari" već tokom 1988. godine angažovali na ... kompjutera, pripremajući čitaoce za sledeći šok.

SLIKA 0101: Compaq DeskPro („Računari 22", januar 1987)

Taman smo naučili šta je to PC i uspeli da sklopimo svoju PC konfiguraciju, snova, kada nas je u „Računarima 22" sačekao trostruki šok: prvi u vidu vesti sa naslovne strane da je PC mrtav i da ga IBM više ne proizvodi. Drugi u vidu opširnog testa savremenog čuda tehnike koje se zvalo Compaq DeskPro 386, verovatno prvog komercijalno raspoloživog računara zasnovanog na mikroprocesoru 80386. Treći u vidu teksta profesora Slavica koji analizira matematičke funkcije IBM-ovog bejzika i pokazuje da, premda je njihova realizacija nešto bolja od realizacije na kućnim računarima, korišćena metoda ne zadovoljava profesionalne zahteve.

Našim čitaocima koji su se upravo vratili iz Minhena sa delovima svog prvog PC-ja naprosto se zavrtelo u glavi: krenuli su sa iluzijom da će konačno makar za njega dati i poslednju paru doći do računara koji je sam vrh tehnologije, a sada vide da su kupili nešto što je toliko zastarelo da se više i ne proizvodi. Srećom, stvari nisu bile tako strašne kao što su izgledale: pokazalo se da će XT računari biti sasvim upotrebljivi i tokom sledećih nekoliko godina. Ali „šok" je bio neophodan – trebalo je „u poslednjem trenutku" objasniti ljudima da investicija u XT tehnologiju nije dobra i da je bolje dati nešto više i kupiti (bar) AT. Koje tako uradio nije se pokajao: mnogi AT računari kupljeni 1987. godine i danas dobro rade i solidno obavljaju neke poslove.

SLIKA 0110: Laser („Računari 30", septembar 1987)

PC-jem se već i u XT danima moglo obaviti dosta posla, ali obaviti posao i zaraditi novac nisu iste stvari: kompjuter sa konstruktorom za pisanje, računanje, vođenje evidencije i tome slično, ali je sve to zvučalo kao olakšanje nečega što bi se i bez kompjutera moglo sasvim lepo obaviti. Otprilike godinu dana posle pristizanja prvih PC klonova u Jugoslaviju konačno će se pojaviti način da se pomoću računara „prave pare" – stono izdavaštvo ili DTP.

Za DTP nije bio dovoljan samo računar – trebalo je obaviti laserski štampač i softver, pa se većina interesenata, možda baš pod uticajem „Računara", opredelila za Hewlett Packard-ov LaserJet SII i paket Ventura Publisher. „Računari" su, naime, u pravom trenutku predstavili LaserJet i njegovu moguću primenu – sledila je veoma zapažena serija tekstova Zorana Životića o stonom izdavaštvu koja je širokoj publici približila ovu oblast toliko da su mnogima počeli da „rade kliker"... Tehnologija koju su koristile mnoge štamparije bila je prevaziđena vremenom, para za investicije u neke visokoprofesionalne foto-slog uređaje malo je ko imao. Štampalo se, izdavalo dosta knjiga... eto prilike! Ubrzo su domaće štamparije zatrpane knjigama koje su pripremljene na PC-ju i odštampane na laseru.

Kolika je sreća u tome, veliko je pitanje – mnogima se danas čini da je izlazak neke knjige objavljene pre „DTP ere" pravi praznik za oči. Problem je u tome što su izdavači često pretervali sa štednjom: preovladao je 300×300 dpi otisak na pausu koji objektivno ne zadovoljava profesionalne standarde, a često su i angažovani ljudi koji nemaju dovoljno znanja ni o tipografiji ni o računarima, ali žele u pokušajima da otplate opremu nude besmisleno niske cene. Nelojalna konkurencija je ubrzo sahranila DTP kao visoko profitabilan posao, ali utisak je ostao: pomoću „tog kompjutera" mogu se zaraditi pare. Taj utisak bio je sasvim dovoljan da PC računari preovladaju domaćim tržištem i da suvereno vladaju njim sve do današnjeg dana.

Ako sam rekao da su se „Računari" u pravo vreme uključili u DTP eru domaće kompjuterske revolucije i da su tu eru na neki način i sami izazvali, drugi način da se pomoću PC računara zarade pare dugo je ostao izvan našeg odmeravanja. Što je, moram da kažem, veoma ozbiljan uređivački propust... zapravo, nije mi jasno kako nam je uspelo da tu primenu tako temeljno zanemarimo, kada je, da tako kažem, „iz našeg faha" – programiranje. Posmatrano iz današnje perspektive, jasno je da su „Računari" programiranju godinama prilazili na, neću reći loš, ali slabo profitabilan način. Trudili smo se naime da objasnimo čitaocima kako da napišu neko svetsko čudo od programa i prodaju ga za velike pare Englezima. Ne kažem da se program nije mogao prodati Englezima, ne kažem da se tako nisu mogle zaraditi vrlo pristojne pare, ali je to posao koji uključuje ogroman procenat rizika, koliko je samo naših programera napisalo sasvim pristojne programe, ali im nije us-pelo da nađu firmu koja bi te remek-dela otkupila, očito nije stvar samo u programiranju nego i u poznavanju tržišta. Sa druge strane, bio je programera koji su razmišljali mnogo „prizemnije": umesto svetskog čuda od programa koje će se prodati u svetu, donet 10 ili 20 hiljada funti, napišimo sasvim običan program koji ćemo uz prethodni dogovor (dakle bez posebnog rizika) prodati komšiji bakalinu za 1000 maraka. Clipper se afirmisao kao idealan jezik za aplikacije tog tipa, a „kliperaši" su tada (a i sada) sasvim pristojno zarađivali – cena od 1000 maraka po programu umerene složenosti održala se do današnjih dana.

U domaćim programerskim krugovima je začudo postojao neki „iskonski prezir" prema onima koji pišu programe u Clipper-u. I taj se ignorantski stav preneo i na stranice „Računara" na kojima Clipper godinama gotovo da nije pomenut. „Računari" će se na pravi način baviti Clipper-om tek mnogo kasnije, u okviru serije tekstova Nenada Batočanina koja je počela u „Računarima 70". Bolje ikad nego nikad, ali trebalo je mnogo ranije!

SLIKA 0111: Školsko zvono („Računari 30-50", „Računari u vašoj školi", april 1988)

„Računari 30" su jedini koji su u ovom osvrtu zaslužili čak dve „slike": važan momenat u njima nije bio samo tekst Zorana Životića o laserskim štampačima, nego i rubrika Nevenka Spalević „Pet plus". Godina 1988. je svakako bila godina uvođenja računarstva u škole, pa je septembar bio pravi trenutak da se i na stranicama „Računara" oglasi „školsko zvono". Školstvo je uvek bilo prilično inertno, pa je i uvođenje računara bilo svojevrstan šok: nastavnicima koji su često „regrutovani" iz redova raznih tehnoloških viškova i pručeni kompjuterima na petnaestodnevnim kursevima trebalo je pomoći da realizuju ambiciozno (premda ne i posebno dobro) zamišljene školske programe.

Kruna rubrike „Pet plus" svakako je bilo specijalno izdanje „Računari u vašoj školi" koje se pojavilo na kioscima aprila 1988. godine. Premda je u nazivu naglašeno „u školi" i premda je značajan prostor posvećen tekstovima pretežno školske tematike, ovo specijalno izdanje nije bilo namenjeno samo školama – to je zapravo bila svojevrsna „repriza" „Računara u vašoj kući", naravno prilagođena vremenu u kome se pojavila. Kroz nekoliko tekstova namenjenih apsolutnim početnicima opisali smo anatomiju personalnog računara, njegove periferijske jedinice, osnove korišćenja i dali pregled raspoloživog komercijalnog softvera.

Repriza „Računara u vašoj kući" nije se završila tekstovima o nabavci i korišćenju računara – predložili smo čitaocima da sami sagrade još jedan računar, ovog puta TIM-011. TIM-011 je konstruisao Nenad Dunjić – Beca u saradnji sa Milanom Tadićem i Ljubišom Gavrilovićem, projekat je otkupio Institut „Mihajlo Pupin" koji je želeo da proizvodi TIM-ove kao školske računare i koji je dobro znao da školski računar ne može uspeti ako ga nema i po kućama, pa je odobrio uslovima da kompletnu tehničku dokumentaciju objave na stranicama „Računara", a Redakciji da organizuje nabavku štampanih kola i drugih komponenti i distribuciju osnovnog softvera, u EPROM-u i na disketi.

Samogradnja TIM-a 011 prošla je dobro, naravno znatno „mirnije" nego svojevremena samogradnja „galaksije" koja je bacila Redakciju u „opsadno stanje". Izučili smo dakle školu samogradnje, a olakšavajuća okolnost bilo je i to što je interesovanje objektivno bilo znatno manje, premda je broj onih koji su uspešno sagradili TIM-011 premašio naša početna očekivanja.

Stoprocentno rasprodati (više nego solidni) tiraž „Računara u vašoj školi" bio je svakako profitabilan za Redakciju, pa se odmah postavilo pitanje eventualnog drugog (trećeg, četvrtog...) specijalnog izdanja toga tipa. Međutim odziv tržišta nije bio dovoljan, pa su „Računari u vašoj školi 2" potpuno pripremljeni ostali neobjavljeni – poslednji nastavak rubrike „Pet plus" objavljen u jubilarnom pedesetom broju našeg časopisa. Posle toga je bilo još par kraćih priloga, ali su „Računari" nekako ostavili školstvo na marginama svojih interesovanja.

SLIKA 1000: Zbogom Spectrum („Računari 38", maj 1988)

Legenda zvana Spectrum se, bar što se „Računara" tiče, završila maja 1988. godine. Izašli su „Računari 38" na naslovnoj strani Ksenija Kepanja, koja se našim čitaocima toliko dopala da su naprosto zahtevali da je vide još jednom, po mogućnosti sa nešto manje odeće (želja im je ispunjena na naslovnoj strani „Računara 40"), u sredini umetak o programu Ventura Publisher, a na ostalim stranicama ni jedan jedini tekst o Spectrumu! Samo par meseci ranije takvo nešto bilo je naprosto nezamislivo: „jaki" tekstovi o Spectrumu i Commodore-u smatrani su okosnicom svakog broja. Ali sve ima svoj početak i svoj kraj, pa tako i Spectrumova „karijera" na stranicama „Računara".

Ako ćemo da govorimo precizno, Spectrumova karijera u „Računarima" nije završena brojem 37... bilo je i kasnije tekstova o njemu, a broj 38 je verovatno osvenuo bez teksta o Spectrumu više nekim sticajem slučajnih okolnosti. Zadnji tekst o Spectrumu je objavljen u „Računarima 49", zvao se „Prozori otvori se" a napisao ga je Žarko Vukosavljević. Međutim, već posle „Računara 38" bilo je jasno da se Spectrumov prozor u „Računarima" neumitno zatvara.

SLIKA 1001: Pirati? Ne, hvala! („Računari 43", oktobar 1988)

Svoj Rubikon na putu metamorfoze u časopis namenjen kompjuterskim profesionalcima „Računari" su prešli oktobra 1988. godine odustajući od piratskih oglasa. Vredi napomenuti da smo određenu distancu prema softverskom piratstvu imali od samoga početka, ali smo ga smatrali za nužno zlo bez koga se u uslovima malih plata i ekonomske krize kompjuteri ne mogu probiti do šire publike. Piratski oglasi su na stranicama našeg časopisa procvetali negde od broja 12, kada je Jelena Rupnik (ponekad zvana i „piratska majka") smislila krilaticu „Obradujte svoj računar – objavite mu oglas u Računarima". Koliko su se Spectrumi i Komodori „radovali" oglasima koji su ih terali da po čitav dan presnimavaju kasete pitanje je za diskusiju, ali je činjenica da je oglasa u kojima se nude igrice a pomalo i ostali programi iz broja u broj bilo sve više.

Piratstvo je, na žalost, ubrzo pokazalo i svoje drugo lice: programi su se dosta bazeli nepotpuni, nije bilo interesa domaći softver u uslovima besplatnog stranog nije imao baš nikakvu šansu, a i sami „Računari" nisu shvatani dovoljno ozbiljno – kako pregovarati sa proizvođačem softvera od koga treba uzeti neki paket na test ako se nekoliko strana iza prikaza kopija njegovog programa nudi za neke smešne pare? A takvi prikazi su iz meseca u mesec postajali sve značajniji za korisničku orijentaciju časopisa „Računari". Led je, obraćajući se direktno stranim proizvođačima softvera, probio Duško Savić koji će sledećih godina objavljivati veoma zapažene prikaze najnovijih verzija programskih paketa, a softver je kasnije pribavljala i Redakcija. Bez piratskih oglasa u listu sve to bi teklo mnogo lakše.

Svi ti (a i mnogi drugi) razlozi govorili su da piratstvu treba reći zbogom, ali se nako vreme nije nalazilo snage da se to uredi. Prelomni trenutak bili su „Računari 43" – uvodnik je naslovljen „Pirati? Ne, hvala!" (aluzija na tada veoma kontraverznu slovenačku parolu „Štafeta? Ne, hvala!"), a sa ostalih stranica nestali su piratski oglasi. Bojazan je izgleda bila neopravdana: „Računari" su dobili na reputaciji a nisu izgubili na čitanosti, tako da se više puta čula i rečenica kojom se danas komentariše i Avramovićev program: „Ako je to bilo tako prosto, zašto zaboga nije mnogo ranije urađeno?" Na to se može odgovoriti samo tvrdnjom da svaka stvar ima neko svoje vreme – možda se od piratskih oglasa moglo odustati onda kada je imalo šta da ih zameni. A zamenili su ih pravi komercijalni oglasi.

Strani kompjuterski (a i većina ostalih) časopisi bukvalno žive od oglasa koji obično zauzimaju 70-80 posto prostora. U domaćoj žurnalistici takvi odnosi nisu uobičajeni: tržište je očito mnogo manje a konkurencija mnogo slabija. Štaviše, ponuda kompjutera je dugo bila znatno manja od potražnje (pre svega zbog problema sa uvozom), pa se sve što je stizalo na tržište i bez posebne reklame prodavalo. „Slika 0110" opisuje dva načina da se na računarima zaradi (DTP i Clipper programiranje), pa je ovo možda prilika da pomenemo treći: prodaja računara. Malobrojni prodavci računara koji su početkom 1980-ih nalazili neka načina da „prekorače" visoko podignutu graničnu rampu bili su u prijatnoj poziciji da drže cene u nebesa: basnoslovno skupe Apple i IBM računare nisu kupovali pojedinci, ali je bilo mnogo velikih društvenih firmi sa dosta para na računu koje su bile spremne da kupuju sve, samo ako je za dinare. Ko je umeo da se snađe mogao je i bez posebne reklame tih godina zgrnuti priličnu paru, a ljudi koji su želeli da kupe kompjuter za sopstvene potrebe morali su ići u Beč ili Minhen, po mogućnosti opet konsultujući reklame u stranim časopisima. Za reklamiranje u „Računarima" kao da nije bilo interesa.

Pa ipak, Sergej Marčenko, koji je 1988. godine preuzeo marketing „Računara", uspeo je da nađe interesenta za oglašavanje, pre svega među velikim firmama koje su želele da utiču u svoj imidž i ilustruju tehnološke pomake koji su činile, ne interesujući se preterano za konkretnu ponudu robe pojedinačnim kupcima. Bila su to vremena kada su fabrike i instituti daleko živeli na ad tržišta nego od velikih

vojnih (i drugih) projekata koji su se dobijali pre svega na osnovu političkog uticaja ali je dobro dolazio i publicitet u medijima – bilo je to dobro tlo za ambiciozne početke komercijalnih oglasa u „Računarima". A krenule su i akcije – najpre u saradnji sa nemačkom firmom „Mraz Electronic" a zatim i sa beogradskom firmom „Jugodata" omogućili smo korisnicima da nabave kompjuter i prateću opremu koja će stići na njihov kućni prag i to pod najpovoljnijim mogućim uslovima.

Prvi oglasi predstavljali su dobru priliku da se Redakcija konačno opremi hardverom neophodnim za kvalitetnu proizvodnju časopisa – prvih pet-šest godina „Računari" su dobrim delom pripremani na opremi koja je bila privatno vlasništvo članova Redakcije. Prvi „zaista naš" AT računar stigao je na sto glavnog urednika početkom 1989. godine, a za njim je došla i druga oprema. Vremenom se skupilo sve što je potrebno ne samo za proizvodnju savremenog računarskog časopisa nego i za ulazak u nove projekte. A najveći takav projekat bio je Sezam.

SLIKA 1010:
Sezam
(„Računari 55", novembar 1989)

Kao što smo čitaoce 1981. godine uveli u svet džepnih računara, 1984. u svet kućnih a 1986. u svet personalaca, kraj 1989. godine doneće prodor u jednu novu oblast, računarske komunikacije. Modemi su u svetu već tada bili veoma popularni, ali kod nas ih je bilo malo ili nimalo. Iz sasvim jasnih razloga: vrednost modema je vrlo mala ako je sve što se njima može učiniti razmena programa sa nekim prijateljem. „Računari" su tokom 1989. godine objavili više tekstova o modemima i modemskim komunikacijama, da bi tokom septembra i oktobra bio osmišljen i realizovan projekat Sezam. Autori ovoga projekta smo **Zoran Životić** i ja, **Dejan Ristanović**, a čitava zamisao bila je (i ostala) veoma ambiciozna: umesto da koristimo neki od postojećih BBS programa, odlučili smo da napišemo softver koji će sjediniti brojne dobre strane raznih sistema sa kojima smo komunicirali, uvodeći novitete koji će olakšati i ubrzati rad. Važna osobina tog programa trebala je da bude pouzdanost – program je morao „snalaziti" u svim mogućim situacijama u skladu sa maksimom „the show must go", čak i ako detektuje neki fatalan problem mora da bude sposoban da nastavi sa radom i signalizira službenom licu šta se tačno desilo. Glavni program napisao je **Zoran Životić**. BBS je pušten u probni rad krajem oktobra, a već početkom novembra je došlo vreme da Sezam, koji smo osnovali zajedno sa **Jovanom Regasekom**, glavnim urednikom „Računara", primi svoje prve korisnike. Zvanični rođendan sistema je 11. novembar, kada je probni period završen.

Iako je Sezam u početku radio na sasvim skromnom hardveru (redakcijski AT i modem) i to isključivo popodne i noću, od prvih dana je predstavljao veliki uspeh: veliki broj korisnika zainteresovao se za njegove usluge, pa smo sve vreme morali da razmišljamo o proširenjima. Originalan, dobro osmišljen i pouzdan softver bio je samo jedan od faktora uspeha – veoma je važno što se sistem orijentisao prema onome što, kako se pokazalo, najviše zanima korisnike, a to su diskusije: napravljena je dobra podela zanimljivih stvari u konferencije (komunikacije, PC, politika, novosti, vicevi...), teme, napisane su uvodna poruke i svakodnevno se radilo na održavanju sistema – korisnici su dobijali brz i kvalifikovan odgovor na pitanja, diskusija je usmeravana, predlozi su se zahvalnošću primani... i uspeh nije izostao!

Nešto pre prvog rođendana Sezam je premostio važnu stepenicu i prešao na komercijalni rad – pokazalo se da je veliki broj korisnika spreman da plati godišnju pretplatu u iznosu od 800 tadašnjih dinara (oko 115 maraka). Ozbiljan rad nije se, naravno, mogao zamisliti na jednoj liniji, pa je komercijalni Sezam najpre radio na tri linije da bi, kada je izabran za zvanični informacioni sistem Šahovske Olimpijade u Novom Sadu, prešao na 13 javnih linija, na Compaqovu opremu. Sezam će kroz sledeće godine rasti, kako po kapacitetima tako i po broju članova i tematici koja je ubrzo nadrasla tematiku našeg časopisa: u centru pažnje su i dalje računarske teme, ali se razgovara i o nauci, kulturi, umetnosti, politici, religiji i mnogim drugim stvarima. Sezam sa svojih 15 javnih linija danas predstavlja najveći javni komunikaciono-informacioni sistem u zemlji i jedan od većih evropskih BBS-ova: ponosni smo što smo ga stvorili.

SLIKA 1011:
Anketa
(„Računari 62", jun 1990)

Godina 1990. je za većinu nas bila po svemu izuzetna: pod dejstvom ekonomskog programa „neprijatelja" Ante Markovića biznis je procvetao, pojavile su se hiljade privatnih preduzeća, od kojih su se neka specijalizovala za uvoz i prodaju kompjutera. Jaka konkurencija je učinila da cene drastično padnu – ubrzo se u Beogradu mogao nabaviti računar po ceni koja nije ništa viša (a često je i niža) od minhenske, i to uz sve blagodeti garantnog roka i servisiranja. „Računari" su ove promene spremno dočekali: konačno su oglasi postali ono što i treba da budu, mesto da se nešto ponudi, objavi cena i razlog zašto treba kupovati tu a ne negde drugde. Od početka 1990. godine pa do današnjeg dana oglasne stranice „Računara" su pune, zahvaljujući pre svega tržišnom ugledu našeg časopisa i trudu koji su kontinuirano ulagali najpre **Esad Jakupović**, a kasnije **Vesna Jeremić**, danas komercijalni urednik „Računara". Naš pogled na tržište tu se nije završavao – iz meseca u mesec **Dejan Veselinović** je prikazivao kompletne konfiguracije i komponenate koje se nude na domaćem tržištu, uporedno ih testirao, analizirao njihove performanse... svako ko je hteo da kupi kompjuter (a takvih nije bilo malo) naprosto nije mogao sebi dopustiti luksuz da pre donošenja odluke ne prelista „Računare"!

Jedan od rezultata tržišne orijentacije „Računara" bila je i velika nagradna igra – čitaoci su dobili Gamin 386 računar i mnoge druge nagrade, a mi smo dobili odgovore na pitanja koja su nas zanimala: ove ankete obuhvatila je 3500 učesnika i rezultirala značajnim rezultatima koji su sumirani u „Računarima 62". To nam je pomoglo da „Računare" usmerimo u pravcu koji najviše zanima čitaoce: pokazalo se pre svega da ogromna većina čitalaca (95%) ima PC kompatibilan računar, što je uslovilo ukidanje „Atari podlistka" i orijentaciju isključivo na kompatibilce. Pokazalo se i da smo negde od „Računara 46" previše „skrenuli" prema korisnicima i zanemarili programere: čitaoci su tražili programerske teme, pa ih je od tada bilo znatno više na stranicama „Računara". Anketa je donela i druge zanimljiva saznanja o domaćoj računarskoj sceni i trendovima njenog razvoja – na osnovu njih početkom 1991. godine smo isplanirali „veliko spremanje" našeg časopisa.

SLIKA 1100:
Novi izgled
(„Računari 71", april 1991)

„Veliko spremanje" je zapravo značilo sasvim novi izgled koji je pretendovao da približi „Računare" modernim evropskim kompjuterskim časopisima: atraktivna oprema, kvalitetan papir, dosta kolor strana, jak oglasni segment... „Računari u novom ruhu" krenuli su na tržište aprila 1991. godine, sa sve novom nagradnom igrom čije su premije bile računari najviše klase i prateći strani softver.

Nije sve u grafičkom izgledu... trebalo je i dalje raditi na sadržaju „Računara", pa smo počeli serije tekstova o Windows programiranju (sam Windows 3.0 predstavljen je u „Računarima 63"), elektronskoj pošti sa svetom, projektom Optima u okviru koga smo za čitaoce formirali nekolike optimalnih računarskih konfiguracija koje se u našim uslovima mogu sklopiti, a važno mesto su zauzele i rubrike nazvane Savetnik. Pomenimo i seriju tekstova **Dragutina Vukovića** o lokalnim mrežama, razgovore sa poznatim piscima koji u radu koriste kompjutere i mnoge druge zanimljive teme.

„Računari 71" su zapravo i po koncepciji i po izgledu slični časopisu koji upravo prelistavate – jedino važno poboljšanje bili su specijalni dodaci koji, počevši od „Računara 78", redovno prate naš časopis. Svaki specijalni dodatak je zapravo knjiga koja ukratko opisuje neki programski jezik ili korisnički program – predstavili smo tako DOS, *Windows*, paskal, C, *Clipper*, *Word*, *WordPerfect*, *AutoCAD*, *Lotus* i mnoge druge programe koji se svakodnevno koriste: za čitaoce je to bila izuzetna prilika da veoma povoljno dođu do literature na našem jeziku prilagođene domaćim potrebama. Izgledalo je da „Računari" ulaze u svoje zlatno doba... a onda je došla kriza.

SLIKA 1101:
Maja Golić
(„Računari 76", novembar 1991)

Možda nije fer, ali za mene (i ne samo za mene) je sinonim za krizu na stranicama „Računara" **Maja Golić**. Ona za to, naravno, nije baš ništa kriva; naprosto je pozirala našem fotografu za naslovnu stranu „Računara 76". Ali, a „Računari 76" su se pokazali kao prilično slab broj: 84 strane umesto ranijih 116, pa još kašnjenja časopisa od preko mesec dana koja je uslovilo da oktobarski broj naprosto bude preimenovan u novembarski. Uz sve to se nekako desilo da u tim „Računarima" ne bude mnogo zanimljivih tekstova; čitaoci su sa razlogom bili nezadovoljni. Pa su uzgred zapamtili i prilično karakteristično ime i prezime manekenke – da li zato što su žalili da je više u garderobi koja bi bila bolje prilagođena prezimenu ili zato što je moderator Sezamove konferencije VICEVI uočio nekakvu paralelu između nje i porinoćja za mija (što je izazvalo pravi urnebes), tek Maja Golić je postala „negativni" heroj „Računara" skoro koliko i pomenute Noge.

A vremena stvarno nisu bila laka – u Hrvatskoj je besneo rat, „Računari" su izgubili više od polovine ranijeg tržišta, sve teže se nabavljao hardver, softver, uvozni papir, lepak i ostali repromaterijal, krenula je inflacija i nemaština... Pa ipak, nekako smo gurali dalje, pre svega zahvaljujući tome što je domaće računarsko tržište tokom čitave ekonomske krize kako tako funkcionisalo i što smo mu se uspešno prilagođavali: koncepcijski smo se okrenuli pre svega profesionalnim korisnicima kompjuterske opreme. Bili smo svesni opasnosti u kojoj se nalazimo, znali smo da će biti potrebno mnogo rada a i pomalo sreće da bi se „Računari" održali na stepenici na koju su se tokom prethodnih godina popeli. Trebalo je preživeti.

I preživeli smo!

SLIKA 1110:
45,000 maraka
(„Računari 92", maj 1993)

„Računari" su se u „punu formu" vratili tokom 1992. i početkom 1993. godine. Redakcija kadrovski ojačana **Vesnom Jeremić** (danas komercijalni urednik) i **Rankom Jovanovic** (danas zamenik glavnog i odgovornog urednika) uspela je da uspostavi punu saradnju sa velikim brojem domaćih računarskih firmi što je, uz ostalo, rezultiralo najvećom nagradnom igrom u istoriji našeg časopisa, sa nagradnim fondom vrednim preko 45.000 maraka. „Računari 92" su se, zajedno sa pratećim specijalnim dodatkom o DOS-u 6.0, Sezamom i nagradama koje će biti dodeljene na konkursu, veoma uspešno predstavili na majskom Sajmu tehnike u Beogradu, a kasnije i na nekim drugim sličnim manifestacijama. Kupovani su, prelistavani, čitani, citirani... kao u najboljim danima, i više od toga!

Na žalost, uzmevraje tema nije značilo da se kriza završila: najviše nas je pogađala njena čisto personalna komponenta, jer je veliki broj naših stalnih saradnika otišao u Kanadu, Australiju i u mnoge druge zemlje kojima izvozimo pamet. Tako smo bili ponuđeni da na stranicama „Računara" objavljujemo dosta prevoda i kompilacija stranih tekstova. Ipak, imam utisak da je kvalitet „Računara", i pored povremenih oscilacija, zadnjih meseci u porastu – objavili smo nekoliko veoma zapaženih serija tekstova, uspeli da pribavimo originalni softver na test i objavili specijalna dodatke koji su naišli na izuzetan prijem kod čitalaca. Najzad, nije malo stvar što se brojevi rasprodaju toliko da se i u samoj Redakciji nedelju ili dve posle izlaska na tržište jedva može naći neki primerak tekućeg broja. Sve to je bilo lepa uvertira za broj koji gledate: jubilarne „Računare 100"!

SLIKA 1111:
"Računari 100"
(maj 1994)

Ova slika je, u danima kada završavamo ovaj tekst, još nenacrtana – zapažaju se samo njeni obrisi u vidu povećanog broja strana, tekstova naših starih saradnika iz zemlje i sveta, specijalnog dodatka o Windows-u, velike nagradne igre u kojoj čitaocima poklanjamo Pentium računar i stotinak drugih vrednih nagrada, opsežnog sajamskog dodatka... Ovako nabrojano, vrlo lepo zvuči; realnost često, kao i uvek, vi procenite.

A što se tiče budućnosti... počeli sam prologom, ali ovaj tekst, Bogu hvala, neće završiti Epilogom! „Računari 100" su važna tačka u životu našeg časopisa, ali posle broja 100 dolazi 101, 102... još mnogo slika koje treba iscrtati u skladu sa vašim interesovanjima i potrebama. Sledićemo koncepciju koja se iskristalisala kroz ovih deset godina, ali ćemo uvek biti spremni na zaokrete koje diktira računarsko tržište. Nadajmo se samo da će situacija u kojoj ćemo pripremati sledeće brojeve našeg časopisa sve manje ličiti na (kad već počeh tekst pominjanjem Lukasa) *Star Wars* a sve više na *Never Ending Story*.

PROŠLOST I TRAČAK BUDUĆNOSTI

Obično se smatra da je duh fleksibilniji od materije, ali u računarstvu važi paradoks da se materija – hardver – razvija brže od duha – softvera. Da su avioni napredovali kao računarski hardver, karta za let bi danas koštala pet dinara. Da je pisanje softvera napredovalo tim tempom, danas bi svi bili čarobnjaci: dovoljna bi bila samo misao o programu, pa da stvari počnu da se dešavaju na ekranu.

Duško Savić

Organizacija programa se može zasnovati na tri načina: na funkcijama, podacima ili objektima. Funkcijski orijentisani programi nastaju tako što programer prvo razrešava one delove programske specifikacije koji su na neki način lakši od ostalih. Naročito su pogodni oni delovi specifikacije u kojima su ulazni podaci, izlazni podaci i način obrade jasni. Tako delovi programa koji su bolje objašnjeni bivaju napisani na početku. Međutim, programer piše funkcije proizvoljno, čim mu postane jasno kako da prevede deo specifikacije. Veze sa ostalim delovima programa nisu definisane, a programer stalno ima osećaj neponovljive originalnosti. Problemi nastaju naročito u fazi održavanja programa, kada konceptualno razdvojene delove treba shvatiti i popravljati kao celinu.

UNUTRAŠNJA STRUKTURA PROGRAMA

U skladu sa ovim stilom, programski jezici nude pojam funkcije kao sredstva za kodiranje algoritama. Svaka funkcija transformiše podatke, izračunavanjem izlaznih podataka u odnosu na ulazne. Problem nije u pojmu funkcije kao takvom već u njihovom povezivanju. U funkcijski orijentisanim programima te veze su labave. Programer ih pravi u ad hoc maniru, prema potrebi. Svesne organizacije programa nema, već program postaje niz snimaka programerovog misaonog procesa. Misli izbijaju slučajno, posle određenog perioda inkubacije, pa je ceo proces nužno nelinearan i nelogički. Zato postaje bitno snimiti stazu kojom se prošlo do rešenja. Dokumentacija postaje lični dnevnik "ova greška popravljena dana ..." I zato programeri mrze pisanje dokumentacije: ono ih podseća (a želim da stvar bude gora, i još pokazuje drugima) koliko su bili neorganizovani u kodiranju.

Funkcija prevashodno računaju, i struktura podataka u njima obično su jednostavne. Kod je što je moguće kraći. Promenljivih ima tačno koliko je potrebno da se obavi posao. Obično postoji jednostavna linearna struktura: ulaz (prenos parametara), proračun, izlaz (vraćanje vrednosti). Rezultat je po pravilu jedna vrednost, obično iz skupa elementarnih podataka jezika.

Programski jezici podržavaju takav način programiranja direktno i čak omogućavaju da se poziv funkcije navede direktno u aritmetičkom izrazu. U C-u je ova ideologija sprovedena do kraja, pa je svaki potprogram funkcija. Kasnije se uvidelo da to ipak nema smisla, pa je pojam funkcije proširen tako da uključuje i potprograme opšteg tipa.

Ako se program zasniva na podacima, njegova organizacija izvire iz potpuno suprotnog smera. Programer prvo analizira rezultate programa i odatle deluje unatrag, prema glavnoj programskoj petlji. Rezultati programa se prirodno dele na dve grupe: one koje program može da izračuna (izlazni rezultati) i na sve druge (ulazni podaci).

Ovaj pristup se može primeniti u svim jezicima. Analiza izlaznih rezultata generiše veliku količinu dokumentacije, pa je faza održavanja olakšana.

Kada se pogleda gotov program, i dalje se koriste potprogrami (među njima i funkcije), ali one sada odgovaraju realnim podacima, i ne zavise od programerove ćudi.

PRINCIP SLIČNOSTI

Unutrašnjost programa treba da bude usklađena sa očekivanjima krajnjeg korisnika. Nove softverske paradigme nastaju kada se očekivanja krajnjeg korisnika masovno promene. Dok se ne pređe sa jedne paradigme na drugu, postoji softverska kriza.

Pogledajmo kako je princip sličnosti delovao kroz istoriju programiranja. Funkcijsko programiranje se obično javlja kada je autor programa identičan korisniku. (Na primer, kada programer piše program kojeg će sam koristiti.) Korisnički interfejs može biti i sasvim loš, ali ništa zato, jer se sve intervencije mogu izvesti direktno u programu. Međutim, šezdesetih godina ovog veka programeri su prestali da budu korisnici. Umesto Besselovih integrala računari su tada počeli da obračunavaju tekuće račune, predviđaju vreme, lansiraju satelite. Pojavili su se korisnici koji nisu programeri. Kao odgovor na to, pojavili su se programeri koji jesu samo to – profesionalni programeri.

Potražnja za programima je rasla eksponencijalno, tako da nije više bilo vremena za bogomdano programiranje. Korisnike ne interesuje programiranje kao autistična umetnost već kao proizvodna usluga. Zatim se pojavilo interaktivno računarstvo, kroz tajmšering (*timesharing*) sisteme. Tu su korisnici mogli sami da zadaju naredbe i van specijalističkih programa. U izvesnom smislu i sami su postajali programeri. Pravi programeri nisu nestali, ali su morali da pređu još jednu stepenicu u evoluciji: od pisanja programa koji obrađuju samo njihove podatke, preko programa koji obrađuju samo podatke drugih ljudi, do programa koji drugima omogućavaju da obrađuju sopstvene podatke. U tehnološkom smislu, unutrašnja organizacija programa menjala se ovako.

Iz prve rečenice principa sličnosti može se zaključiti: **Objektno programiranje je rešenje za interaktivno programiranje.**

Po definiciji, objekti u sebi uključuju i podatke i relevantne operacije. Sve dok se još neki veliki zaokret u korisničkim potrebama ne dogodi, objektno programiranje će biti najbolji izbor za interaktivne aplikacije.

JEZICI NEKAD I SAD

Razvojni put programskih jezika od fortrana iz 1954-te godine do danas je uglavnom poznat. Ovde ćemo se baviti promenama u poslednjih desetak godina.

Osnovna karakteristika kućnih i ličnih računara je programabilnost. U prvo vreme (1975. godina) samo to se i moglo raditi sa tim mašinama. (I danas ćete sresti ljude koji će vam reći: "Ja ništa ne znam sa računarom, ja ga samo koristim," izvinjavajući se što ne znaju da programiraju.) Osnovni jezik je bio bejzik, pod pomognut mašinskim jezikom, odnosno asemblerom. Prvi veliki susret klasičnog programiranja i kućnih računara dogodio se kroz paskal. To je bio "pravi" jezik, koji je ipak mogao da stane u 32, 48 ili (neverovatnih) 64 kilobajta. Na komodoru i spektrumu bilo je i drugih pokušaja, kao što su fortran i fort. Ipak, najviše se "trošio" mašinski jezik, zbog razbijanja zaštite. Otuda i velika popularnost raznih monitora i disasemblera, i silni članci sa uputstvima po računarskim časopisima. Zatim je došlo vreme PC standarda, od 1981. godine. Masovnost i neograničeni rast nagnali su pisce svih postojećih programskih jezika i sistema da se prebace na PC. Bukvalno, svaki programski jezik od značaja je bio prepravljen, uključujući čak i PL/I. Ali, PC je bio tretiran kao "veliki" računar elonih dimenzija, pa se i način rada sa dotadašnjih "velikih" sistema preneo po automatizmu. Razvoj programa je bio izrazito mučan. Prvo vam je bio potreban neki tekst editor u kome bi unosili tekst programa, a zatim bi napustili editor, iz DOS-a pokrenuli prevodilac, sačekali izveštaj o greškama, zatim ušli u editor da ih pročitate, zatim učitali tekst programa da biste ispravili greške, pa opet isto to. Kada prođe sintaksna kontrola, pozvali biste novi program, linker, koji bi prevedeno spajao sa postojećim bibliotekama na disketi (diskovi praktično još nisu ni postojali), sačekali njegov izveštaj o greškama, pa onda u editor da pročitate to, pa u tekst programa da ispravite pa eventualne nove sintaksne greške, pa opet u linker, i sve to pre nego što program uopšte nešto počne da radi. A kad proradi i ustanovite greške u izvršavanju programa, opet Jovo nanovo. Čak i kasnije na XT-u sa tvrdim (ali sporim) diskom automatizovano prevođenje programa u MS FORTRAN-u je trajalo pola sata.

Produktivnost – nikakva, ali za bolje se nije znalo. A onda, 1983, pojavio se Filip Kan, Francuz, i svojim galskim duhom ukazao na suštinu stvari: na interaktivnom računaru i razvoj programa mora biti interaktivan. Ali odatle ne sledi da taj interaktivan rad mora biti sa interaktivnim jezikom. On se odlučio za paskal: *Turbo Pascal 1.0*, program koji je spojio kvalitete prevodioca i interpretera. To jeste prevodilac, jer se dobija program na mašinskom jeziku, i izvršava se desetinama puta brže od bejzika. Ali interpreteri su pogodni za razvoj programa jer je traženje grešaka olakšano. U turbo-paskalu to je simulirano time što se u slučaju greške kursor automatski postavljao na odgovarajuće mesto u tekstu programa. I slično bejziku, nije bilo začaranog ciklusa editor – linker – greške, već se sve događalo mnogo brže. Programeri su to prihvatili oberučke i danas, posle deset godina, nema nijednog programskog jezika koji nema integrisani editor.

To ne znači da su interpreteri nestali. Bej-

zik je preživeo, i jedno vreme bio poprište žestoke konkurencije između Borlanda i Majkrosofta. Borland je kasnije otpao, ali drugi se povremeno ubacuju pa u svakom trenutku postoje dva-tri bejzika na tržištu. Javljaju se i drugi jezici. Fort je stalno prisutan ali nikakvu veliku popularnost nije zadobio. U nedostatku jedne velike kompanije koja bi stala iza njega, fort je ostao arhaičan. Danas je besmisleno da vam se nudi da program pišete u delićima dužine od jednog kilobajta samo zato jer su tako radili fortovi na prvim mikroprocesorima. To je šteta, jer u fort su bile polagane ogromne nade: štedljivost memorije, velika brzina izvršavanja, mogućnost samoproširivanja, rad u realnom vremenu, direktan pristup svim delovima računara.

Slična je sudbina REXX-a, jezika koji je na nekim računarima čak i osnovni jezik. REXX jeste neka vrsta bejzika, a naročito je pogodan za automatizaciju rada na računaru, dakle nešto kao bejzik za *batch* obrade. Iako u jednom IBM-ovom centru postoji preko milion linija napisanih na ovom jeziku, masovne popularnosti nema.

Svojevremeno se mnogo buke dizalo i oko prologa, jezik veštačke inteligencije, a jednog dana svi će programi postati ekspertski sistemi. Takve nade je naročito potpirivala činjenica da je Borland 1987. izbacio na tržište *Turbo Prolog*. Nade su pokopane godinu-dve kasnije, kada ga je Borland izbacio iz proizvodnog programa i skoncentrisao se na *Turbo Pascal* i *Turbo C/C++*.

TURBO PASCAL

Zbog ogromne popularnosti turbo-paskala na njemu je najbolje pratiti promene koje je programerski svet doživeo za ovih desetak godina. Dva su glavna faktora uspona tog i drugih jezika: poboljšani hardver sa jedne strane i poboljšanje organizacija programiranja sa druge strane. Diskovi su postali veći i desetinama puta brži, procesori stotinama puta brži, memorije se proširile od 640K do 4, 8 ili 16 megabajta. Takođe, povećali su se rezolucije i broj boja grafičkih sistema. To je omogućilo da potpuno nove klase jezika dođu na svetlost dana.

Drugi faktor – povećanje radne snage koja učestvuje u proizvodnji programskih jezika – takođe se promenio. *Turbo Pascal* je nastao kao proizvod jednog čoveka za CP/M računare, a danas se njime bavi čitav tim ljudi. U Majkrosoftu je sve to još veće i vidljivije, pa tu sad radi preko deset hiljada ljudi (a broj se i dalje povećava). Kao rezultat, programi su postali veći po prostoru koji im je potreban na disku, po broju disketa na kojima se isporučuju, po hiljadama strana dokumentacije, po broju urađenih primera, a uvedena je i interaktivna dokumentacija (*help*).

U programerskom pogledu, prihvaćene su sve postavke objektnog programiranja: modularnost, lokalizacija i zaštita podataka, nasleđivanje, polimorfizam itd. To se naročito drastično vidi na primeru turbo-paskala. U 1986. godini, u verziji 3.0, to je još uvek pravi paskal tačno onako kako ga je zamislio Virt. Program je obavezno monolitan, sa velikim brojem procedura i funkcija, ali se one svaki put moraju prevoditi. Editor koji je tri godine ranije izgledao kao "čudo nauke i tehnike" sada je skromnih mogućnosti, a do memorije preko 64K moglo se doći samo preko pokazivača. Postojale su dopunske biblioteke za grafiku, baze podataka i slično, ali i one su postajale samo deo jednog velikog programa. Ukratko, paskal doveden do savršenstva, ali neadekvatan tadašnjem hardveru.

VELIKI SKOK

Zato je verzija 4.0 bila veliki skok, naročito u semantičkom smislu. Uvedeni su moduli (*units*) koji su omogućavali zasebno prevođenje. Dokumentacija je porasla, naredbe za grafiku su postale standardne. Verzija 5.0 nije donela suštinske novine, ali verzija 5.5 iz 1990. godine i te kako jeste: uvedeni su objekti. U tom trenutku bilo je dva miliona registrovanih korisnika turbo-paskala i svi su oni odjednom mogli da se uključe u najnovija trendove. Ali, osim sintaksne podrške druge vrste nije bilo. Objekti su divni, ali najbolje deluju kroz nasleđivanje, kada programer preuzima već postojeće objekte i malo ih preinačuje u svoju korist. Zato je ubrzo usledila verzija 6.0. Osim poboljšanog editora (rad sa više prozora), glavno poboljšanje je *Turbo Vision*, sistem objekata za korisnički interfejs pod DOS-om. Tu se objekti iskazuju u punoj snazi, koliko ne-grafički izgled DOS-a to dopušta: svaka aplikacija napisana u turbovizij ima meni, reaguje na miša, podržava više prozora na ekranu, razne vrste dijaloga sa korisnikom itd. Problem tu iskrsava na potpuno neočekivanoj strani: *turbovizija* je jezik za sebe, a samo koristi sintaksu paskala i objekata. Ali to je problem kojeg svi koji žele da koriste objekte moraju da prevaziđu: da osim "domaćeg" jezika nauče još i "podjezike" za prozore, baze podataka, matematičke operacije itd. Poslednjeg dana 1992. godine pojavila se i danas aktuelna verzija *Turbo Pascal 7.0*. Obuhvata tri razne hardverske podgrupe modernih PC računara: DOS "ispod" 640K, DOS "iznad" 640K i *Windows 3.0* i *3.1*. I dalje je, dakle, moguće praviti programe za XT računare ali takođe i za one sa boljim procesorima i megabajtima memorije. Uz *turbo*-paskal se isporučuje DOS *extender*, što znači da može da radi u zaštićenom režimu procesora 286 i viših. Konkretno, program može biti dugačak 550K a da ipak bude u stanju da iskoristi dodatnih 1.5MB, ako ih ima. Jedini problem koji od prvih dana PC-ja nije razrešen je ograničenje na dužinu od 64KB po jednom nizu ili segmentu (mada, naravno, postoje načini da se i to prevaziđe). Za DOS je i u verziji 7.0 zadužena turbovizija, dok je pod *Windows*-ima život sasvim drukčiji!

Pod *Windows*-ima se koristi potpuno druga hijerarhija objekata, a programiranje počinje ne u editoru jezika nego u editoru korisničkog interfejsa. Dakle, prvo napravite menije prozora, definišete procedure koje će odgovarati na svaku pojedinu opciju menija – i to je to! Napravili ste pravi program pod *Windows*-ima! U izvesnom smislu to je čak mnogo jednostavnije nego pod DOS-om. A o lakoći korišćenja da i ne govorimo.

Tu vidimo interaktivnost na delu: programeri interaktivno pišu interaktivne programe. Trenuci "čekanja na prevodilac" svedeni su na minimum.

Sličan je razvojni put i ostalih jezika koji su preživeli do današnjeg dana. Mnogo ih je i otpalo: u jednom trenutku bilo je čak petnaestak proizvođača C jezika, danas ih je znatno manje. Nije više dovoljno imati jezik, nego i editor i dibager i profajler, i asembler i klase... Dani kada je jedna osoba mogla da napiše C nepovratno su prošli. Preciznije, jedna osoba danas ne može da napiše C++, ali sa njim može da napiše potpuno profesionalan program, koji će do maksimuma odgovoriti očekivanjima modernih korisnika.

PROTOTIPOVI

Jezici pod *Windows*-ima, ali i na svim drugim platformama, sve više navode na razvoj programa pomoću prototipova. Ideja je da se prvo napravi prototip – maketa programa, i da se ponudi na ocenu budućem korisniku. Obično se napravi korisnički interfejs: meniji, polja na ekranu, izgled tipskih rezultata, aktivnost tastature i miša. Još uvek rutine za snimanje na disk ili štampanje ne rade (da rade, to bi bio skoro gotov program). Povratna informacija dobijena kroz testiranje prototipa je dragocena. Prvo se vidi da li je korisnik uopšte tražio takav program, koji podaci i kontrole nedostaju, kako treba da izgleda rezultat, koji su tasteri zgodni a koji nisu. Kada se utvrdi da će budući program biti od koristi, pristupa se dovršavanju tj. kodiranju obrade i proračuna. Faze dizajniranja i testiranja dizajna (za razliku od testiranja programa) se tako spajaju u jednu. Ranije se testiranje dizajna programa sprovodilo jedino kroz testiranje napisanog programa. Ako bi se ustanovilo da dizajn ne valja, čitave delove programa bi trebalo menjati ili pisati iznova. Uz pomoć prototipova ustanovljavaju se iste te greške a da se program uopšte i ne piše.

Sada se sve zgodno uklapa. Spolja se vidi interfejs, a on se sastoji iz prozora, ikona, senzitivnih delova ekrana, dijaloga i sl. Iznutra, svakom tom elementu odgovara po jedan objekat. Ono što može korisnik istovetno je sa onim što program iznutra može. Zadovoljen je princip sličnosti koji insistira da razlike između programa, interfejsa i korisnikovih potreba budu minimizovane.

NE, NE, NIJE KRAJ

I tako, sleduje nam da učimo objekte i njihove hijerarhije? Možda da, možda ne. Već se sada tračci budućnosti naziru kroz prozore. Jednog dana razvojna sredina na PC-ju i drugim platformama biće toliko bogata da će programer (ili će to možda biti običan korisnik!?), kao što u šetnji predivnim vrtom bira koji će cvet pomirisati, birati iz kojih delova želi da mu se sastoji program, i da se program onda sam od sebe "linkuje" i proradi? Programi kao što su *ObjectVision*, *Visual BASIC* i mnogi drugi nas uveravaju da je to moguće. No, kao i uvek, najlepši cvet je iza sledećeg ugla. U međuvremenu, daleko smo dogurali. Počeli smo sa bejzikom a sada...

Ko obrađuje čije podatke	*Stil kodiranja*
programi koji obrađuju programerova podatke	funkcijsko programiranje
programi koji obrađuju korisnikove podatke	programiranje koje polazi od izlaznih podataka
programi koji omogućavaju korisniku da sam obrađuje svoje podatke (interaktivan rad)	objektno programiranje

SVEST O NOVELU

Iako se broj ozbiljnih svetskih softverskih firmi meri stotinama a broj programa verovatno stotinama hiljada, svega tri proizvoda se po svojoj masovnosti primene izdvajaju iz ove gomile – DOS operativni sistem, *Windows* grafički orijentisan operativni sistem i *Novell Netware* mrežni operativni sistem. Do sada su se u „Računarima" programerske teme gotovo isključivo odnosile na prvi, nešto manje na drugi, dok o *Netware*-u skoro da uopšte nije bilo reči.

Zoran Životić

Korišćenje mogućnosti *Netware* operativnog sistema u velikoj meri je zapostavljeno iz dva razloga. Prvi je potreba nabavke dodatnih alata, odnosno biblioteka kojima se ostvaruje pristup pojedinim funkcijama. Ono što se u suštini dobija je zapravo dokumentacija – sam kod kojim se pojedine funkcije izvršavaju je više nego jednostavan, ali su strukture podataka koji se razmenjuju sa operativnim sistemom često složene, pa se bez dokumentacije ne vredi ni upuštati u posao. Drugi razlog leži u činjenici da se „Novel" politeno potrudio da postojeći programi mogu sasvim lepo da rade u mreži čak i ako toga nisu svesni. Programi koji koriste, recimo, zaključavanje slogova oslonjeno na DOS-ov SHARE radiće sasvim korektno i sa datotekama na mrežnom disku. Štampanje na lokalnom štampaču čak i preko BIOS funkcija se može preusmeriti na mrežni štampač bez ikakve izmene aplikacije.

Ima li onda uopšte smisla komplikovati program nečim što možda u praksi i neće dobiti priliku da se iskaže ukoliko se aplikacija nikada ne „zavrti" na mreži? Mislim da je odgovor – svakako da. Program iz kategorije „poslovnih aplikacija" danas ima velike šanse da radi u mrežnom okruženju i tada će korisnicima i te kako dobro doći dodatni komfor koji će imati ukoliko vaša aplikacija ume da iskoristi mogućnosti mrežnog operativnog sistema.

Puno razumevanje primera koji će ovde biti dati zahteva dosta prethodnih teorijskih objašnjenja o organizaciji *Novell Netware*-a pa i dosta nabrajanja opcija pojedinih funkcija. Ipak, svrha ovog teksta je drugačije prirode – namera je da se na dva praktična primera koje se direktno mogu iskoristiti u vašim aplikacijama pokaže kako to izgleda u praksi. Zato će tekst neminovno ostaviti mnogo otvorenih pitanjam, ali verujem da će ipak dati dovoljno osnove da možete i sami nastaviti sa probama. Da iskoristim odmah priliku da se zahvalim Bojanu Pavkoviću koji mi je pružio deo informacija potrebnih za ovaj tekst.

Način na koji se izvode *Netware* specifične funkcije ni po čemu se ne razlikuje od onog na šta smo navikli pri direktnom korišćenju DOS funkcije, čak je i interapt isti. Dakle, registrom AX identifikujemo funkciju, ostale registre napunimo potrebnim parametrima i pozovemo INT 21h. Novel je nedavno objavio da će ubuduće, zbog konflikata koji su se pojavili sa nekim drugim proizvodima, dokumentovati pozive koji idu preko takozvanog *multiplex* interapta 2F koji i jeste legalan metod nadogradnje DOS-a (isti je način pripreme registara, samo se int 21h zamenjuje sa int 2f). Ipak, int 21h pozivi će ostati neograničeno dugo podržani, tako da praktično nema razlike koji od ova dva interapta koristite.

Jedina specifičnost *Netware*-a je što sami registri u najvećem broju slučajeva nisu dovoljni da se prenesu svi potrebni parametri, pa se koristi metod ulazno/izlaznih paketa. Rezervišemo memoriju za izlazni paket i paket popunimo u zavisnosti od konkretne funkcije, rezervišemo memoriju za povratni paket i zatim adrese ova dva paketa prosleđujemo funkciji. Ovaj metod zapravo ilustruje i način na koji se samo izvršavanje funkcije odvija. Vaš paket se zahtevom se retko obrađuje na samoj radnoj stanici, obično se šalje serveru – server obrađuje zahtev, popunjava povratni paket i vraća ga radnoj stanici.

S obzirom da svaka funkcija (odnosno grupa funkcija) zahteva različitu strukturu paketa

Slika 1



Preuzimanje poruke konzole

u praksi se koriste dva metoda njihovog formiranja. Prvi je da se jednostavno odvoji bafer dovoljne dužine a da se pojedina polja paketa popunjavanju „gađanjem" odgovarajućeg ofseta. Drugi je da se sve potrebne strukture mapiraju preko klasičnih deklaracija strukture – metod daleko pregledniji ali i zamorniji jer zahteva dosta pisanja. U primerima će ipak on biti korišćen.

Da se vratimo na konkretan problem. Na slici 1 su date dve funkcije kojima možemo kontrolisati protok poruka na konzoli. Prva funkcija **cast(int state)** kontroliše ponašanje samog NET SHELL-a – ukoliko parametar **state** postavimo na 3 biće potpuno sprečeno „iskakanje" poruka u poslednjem redu ekrana. Vrednost 0 vraća potpunu kontrolu poruka NET u.

Kada jednom sprečite prikaz ovih poruka vašem programu ostaje mogućnost da ih sam preuzima. S obzirom da se konzolne poruke ne mogu taložiti na serveru već su ograničene samo na jednu poruku koja će beskonačno čekati (i sprečiti prijem novih poruka) da je preuzmete, potrebno je u programu obezbediti da se u nekim vremenskim intervalima obavlja upit serveru i preuzimanje poruke. Funkcija koja obavlja ovaj posao je **int PoolMessage(char *msg)** i ilustruje tehnije pomenuti metod pripreme parametara u paketima. Paket sa zahtevom **(S_PoolMessage)** u ovom slučaju je veoma jednostavan – sadrži svega dva polja koja su inače standardna u svim slučajevima. U prvo polje **length** se upisuje dužina paketa a zatim sledi polje **function** kojim se zapravo identifikuje funkcija koju želimo da izvršimo. I povratni paket **(R_PoolMessage)** je takođe jednostavan – obavezno polje **length** određuje dužinu paketa koju smo u stanju da primimo (u zavisnosti od konkretne funkcije propisana je i minimalna dužina paketa), dok će sledeća dva polja – dužina poruke i tekst poruke popuniti server.

Ima više načina primene funkcija sa slike 1. Jedan je da se preuzme tajmer interapt i da se u regularnim vremenskim intervalima vrši poziv funkcije **PoolMessage()**. Drugi, jednostavniji i često dovoljno primenljiv, može biti ugradnja ovog poziva u centralizovanu rutinu za čitanje tastature. U oba slučaja primljene poruke mogu biti slagane u interni niz programa i prikazane korisniku bilo odmah bilo u nekom pogodnom trenutku kada se po toku programa pretpostavlja da korisnik nije mnogo angažovan u radu. S obzirom da se poruke primaju u standardnom formatu iz koga se može „izvaditi" ime korisnika koji je poruku poslao, moguća su i dalja poboljšanja. Tako na primer, sistemske poruke (ili samo neke od njih, poput obaveštenja o spuštanju servera) mogu da se prikažu odmah, a ostale tek kada je korisnik slobodniji.

Pri izradi aplikacije treba voditi računa o dve stvari. Jedna je da se po izlasku iz programa vrati prethodno stanje načina prikaza poruke, kako korisnik ne bi trajno ostao bez njih

BEEP, BEEP, PORUKA

Da li vas je ikada iznerviralo ono čuveno *beep* praćeno porukom u poslednjem redu ekrana, kojom vas neki drugi korisnik (ili server) na mreži obaveštava o nečem „važnom" i koja je u stanju da iskoči na ekran bez obzira na to šta vi u tom trenutku radite? Taman se koncentrišete na tekst koji pišete, kucate sve brže i brže, kad odjednom ... *beep PERA: Hi! Šta radiš?* i sve se zamrzne dok ne pritisnete Ctrl Enter. Poruke se naravno mogu isključiti (CAST naredbom) ali među njima ima i onih sistemske prirode koje ipak ne bi trebalo propustiti. Moguće je zadržati prikaz samo sistemskih poruka, ali ne bi bilo loše da ipak u nekom trenutku korisnik može da pogleda da li je bilo i onih manje važnih. Ako želite da prijatno iznenadite korisnika vašeg programa, eto sitnice koja se lako može unaprediti u odnosu na ono što standardno nudi NET SHELL.

Slika 2

Direktno štampanje u PRINT QUE

Drugi je da upite o postajanju poruka na serveru ne treba vršiti često. Ukoliko to radite bez ikakve pauze (dok recimo čekate da korisnik pritisne neki taster), strahovito ćete povećati promet i možete potpuno zagušiti mrežu. Interval provera kraći od jedne sekunde pa i više od toga retko ima smisla.

ŠTAMPA BEZ POSREDNIKA

Iako se štampanje, zahvaljujući programu CAPTURE, može izvesti na mrežnom štampaču bez modifikovanja aplikacije, komfor koji se tako pruža korisniku nije maksimalan. Ukoliko našto ne preduzmete, korisnik će nepotrebno čekati da nakon postale štampe ona zaista bude pokrenuta na štampaču, a ukoliko se predomisli i poželi da štampu prekine, moraće da izlezi iz aplikacije i pokrene PCONSOLE program kako bi, kako se to popularno kaže, ubio štampu. Iako se oba problema mogu rešiti i bez poznavanja Novelovih funkcija, najelegantnije rešenje se može dobiti samo ako se koristi direktno štampanje u odgovarajući PRINT QUE – posao koji se može obaviti funkcijama sa slike 2.

Iako slika 2, na prvi pogled deluje veoma komplikovano, radi se isključivo o posledici namere da se dugačkim imenima polja struktura i promenljivih obezbedi lakša identifikacija velikog broj parametara. U osnovi, procedura štampanja direktno u QUE je veoma jednostavna. Prvo je potrebno, na osnovu naziva QUE a, odrediti interni identifikacioni broj tog reda (**GetBinderyObjectID**). Zatim se poziva funkcija za otvaranje novog ulaza u redu čekanja (**CreateQueJob**) i nakon toga sledi jednostavno ispisivanje sadržaja štampe u datoteku predefinisanog naziva NETQ. Nakon što se pošalje poslednji bajt i datoteka NETQ zatvori, ostaje da funkcijom **StartQueJob** završimo posao, a server zatim preuzima stvar u svoje ruke. Komfor koji se korisniku nudi ovakvim načinom štampanja ne ogleda se samo u tome što će štampanje (naravno, ukoliko neki drugi posao nije u toku) počeli odmah, što rezidentni CAPTURE postaje nepotreban, već i u takvim sitnicama kao što je mogućnost da štampi dodelite opis (polje **Job.TextJobDescription**) koji će se pojaviti pri pregledu reda programom PCONSOLE (na primer „Faktura 512/94", „Nalog za knjiženje 123" itd).

Očigledno je, dakle, da je procedura prilično jednostavna. Ono što čini celu priču komplikovanom je poznavanje struktura paketa i njihovo korektno popunjavanje. Tako bi priča o sadržaju polja **ClientRecordArea** (koji je i sam struktura!) strukture **Job** zahtevala bar još jednu stranu teksta – ovde je samo mislenozno u treći bajt upisan broj jedan, koji označava da želimo štampu u jednom primerku. U ostalim bajtovima se krije niz polja kojima se određuje način štampanja (kada iz PCONSOLE programa zatržite širu informaciju o nekom poslu, dobićete približnu sliku o kakvim parametrima se radi).

Od ove tačke do potpune kontrole reda za štampanje nije dalek put. Sa svega nekoliko novih funkcija dobijamo mogućnost pregleda raspoloživih redova za štampu (PRINT QUE), pregled njihovog trenutnog sadržaja, izbacivanje nekog od poslova koji čekaju u redu za štampu itd. Na žalost, svaka od njih zahteva opet svoje strukture paketa, što bi sve zajedno odnelo suviše prostora. Na slici 3 je ipak data jedna od važnih funkcija kojom možemo dobiti listu objekata. U objekte spadaju i redovi za štampu (ObjectType = 3), korisnici mreže (ObjectType = 2) itd. U centralnoj tački ove funkcije je petlja čiji je sadržaj ostavljen otvoren – funkcija vraća listu vrednosti jednu po jednu, pa je na vama da odlučite šta sa njima – da li odmah da budu prikazane korisniku ili da se slažu u interni niz radi dalje obrade.

Slika 3

Lista objekata

Slika 4

Pomoćne funkcije i makroi

VIŠI BAJT - NIŽI BAJT

Na kraju samo još jedna napomena. Novel ze sve celobrojne promenljive (int i long) koristi redosled „viši bajt – niži bajt", što je suprotno prirodi Intelove arhitekture. Ipak, sam sadržaj promenljive je retko interesantan, pa nema potrebe vršiti konverziju. Tipičan primer je ID PRINT QUE a koji se dobija funkcijom **GetBinderyObjectID** – on je parametar drugih funkcija, a njegova vrednost nema nikakvog značaja za program. U primerima se zato konverzija koristi samo kada je to neophodno. Ukoliko poželite da pregledate realan sadržaj neke promenljive, moraćete da dodate konverziju formata. Funkcije za ovu namenu su date na slici 4. U primerima je zbog uštede prostora potpuno izostavljena provera greški – ona se lako može ugraditi ispitivanjem vrednosti koju vraća funkcija **_ShellRequest**.

Nadam se da će funkcije iz ovog teksta pronaći mesto u vašim programima i da će i ove skromna upusva bib dovoljna da nastavite sa probama i učinite da vaš program korisniku ponudi više.

TEK JE DVANAEST SATI

ili kako za pet sati napisati program koji će se naći na 70% računara u zemlji.

Dejan Jelović

U trenutku kada ovo čitate verovatno je u toku sajam kompjutera. Ukoliko ste programer preporučujem vam da ga obavezno posetite. Microsoft i Borland mogu da imaju najbolje mašine i najbolje alate, ali ono gde vi možete da ih preteknete jesu dobre ideje. A koje je bolje mesto za dobijanje revolucionarnih ideja od hiljada kvadratnih metara prepunjenih kompjuterima? Kada smo kod dobrih ideja, pre nekih mesec dana pustio sam u svet kao *freeware* jedan *screen saver* za *Windows*. Program se zove *Company Logo* i jedina mu je funkcija da rotira neki string na ekranu. Sad da li ćete za taj string da upišete svoje ime, ime svoje devojke ili ime firme (što je bila osnovna ideja) nije ni bitno. Bitno je da se *Company Logo* polako širi po računarima širom Jugoslavije. Već sam video po nekim firmama kako na ekranima rotira njihove imene.

Moj plan jeste da sačekam da se *Company Logo* dovoljno raširi da bih mogao da ga koristim kao besplatnu reklamu. Moći ću recimo da kažem: „Naši programi se koriste u 70% preduzeća u Jugoslaviji." Ili zamislite samo: sedite preko puta nekoga ko razmišlja da li da vam poveri važan posao. On vas, kako je uobičajeno, zamoli da mu pokažete neki program koji ste uradili kako bi proverio vaše znanje, a vi ćete: „Nema potrebe za tim. Naših programa ima svuda. Eto taj *screen saver* koji se sada vrti na vašem ekranu je mojih ruku delo." Moćno, a?

ŠTA ZNA DETE ŠTA JE TONA

Najlepša stvar jeste to što mi je da bih napisao *Company Logo screen saver* trebalo nekih pet sati čistog pisanja i oko sat vremena sitnog doterivanja u skladu sa zamerkama onih koji su testirali beta verziju.

Kada sam poslednji put bio kod Mijana, obojica smo ispljuvali *Program Manager*. Najviše nam je kao programerima smetalo što ne postoji lak način da se pri startovanju programa izabere početni direktorijum. Ukoliko imate više projekata na kojima istovremeno radite, stvar ume da bude veoma naporna. Osim toga, ukoliko ste u nekoj aplikaciji i želite da pokrenete još neki program, imate dva izbora: ili ćete za *Program Manager* da rezervišete neko parče ekrana i time smanjite i tako malu raspoloživu površinu, ili ćete beso mučno da stiskate Alt Tab dok do njega ne dođete. Razmišljajući o svemu tome na putu kući, palo mi je na pamet da napravim malu zamenu za *Program Manager*. Imao sam nekih četiri sata do sledeće obaveze, sasvim dovoljno da završim posao.

Listing 4

```
Void TProgListBox  WMKeyDown (TMessage &Msg)
{
        int index = GetSelIndex ();  clan
        if (index >= 0)
                clan = (WORD)SendMessage (HWindow, LB_GETITEMDATA, index, 0);
        int i,
        switch (Msg.WParam) {
                case VK_INSERT  counter = 1
                                                        for (i = 0; i < programs; i++)
                                                            if (progres
[i] notdeleted)
                                                                break,
                                                        if (i < programs &&
GetApplication () ->ExecDialog (new TProgItemAdd (Parent, i)) != IDCANCEL) {
                                                            int k = AddString
(progres [i] name)
                                                            SendMessage (HWindow,
LB_SETITEMDATA, k, i);
                                                            SetSelIndex (k)
                                                            progres [i] notdeleted
= TRUE
                        }
                                                        break,
                case VK_DELETE  if (index >= 0) {
                                                            char s [150]
                                                            char q [MAXPATH],
                                                            GetSelString (q,
MAXPATH)
                                                            *s = 0,
                                                            counter = 1,
                                                            wsprintf (s, "Are you
sure you want to delete item \ %s\ ?" q),
                                                            if (MessageBox
(Parent->HWindow, s, "Delete", MB_YESNO | MB_ICONEXCLAMATION) == IDYES) {
                                                                DeleteDir
(progres (clan) directory next);
                                                                progres
(clan) notdeleted = FALSE
                                        DeleteString (index)
                                                            }
                                }
                                                break
                case VK_ESCAPE  Parent->CloseWindow (),
                                                return,
                case VK_RETURN  RunProgres (index, clan)
                                                return,
                default                 DefWndProc (Msg),
        }
        UpdateEdit (),
```

Promene *default* ponašanja List box-a

Listing 3

```
void TProgEdit  WMLButtonDown (TMessage &Msg)
{
        DefWndProc (Msg),
        Msg.Message = WM_LBUTTONUP
        DefWndProc (Msg);
        SendMessage (Parent->HWindow, WM_SYSCOMMAND, SC_MOVE | HTCAPTION, 0L),
}
```

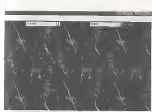
Kako pomerato prozor bez *titlebar*-a

Listing 2

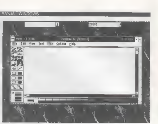
```
void TProgListBox  WMSysKeydown (TMessage &Msg)
{
        if (Msg.WParam == VK_RETURN) {
                int index = GetSelIndex ();  clan,
                if (index >= 0)
                        clan = (WORD)SendMessage (HWindow, LB_GETITEMDATA
index, 0)
                if (index >= 0) {
                        counter = 1
                        if (GetApplication () ->ExecDialog (new TProgItemEdit
(Parent, clan)) != IDCANCEL) {
                                DeleteString (index),
                                int t = AddString (progres (clan) name)
                                SendMessage (HWindow  LB_SETITEMDATA  t
clan),
                                SetSelIndex (t)
                        }
                }
        }
        UpdateEdit ()
}
```

Listing 4

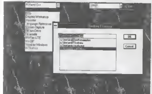
```
void TProgWindow  RunProgram ()
{
        char cmd [CMDLENGTH]
        cmdline->GetText (cmd, CMDLENGTH)
        if (*cmd) {
                char *space = strchr (cmd    )
                if (space)
                        *space = 0
                int ustev = strchr (cmd  * ) || strch  (cmd   ?
                if (space)
                        *space =
                if (ustev) {
                        if (GetApplication () ->ExecDialog (new TFileDialog
(this, SD_FILEOPEN  cmd)) == IDOK)
                                cmdline->SetText (cmd)
                        return,
                }
                if (WinExec (cmd, SW_SHOW) < 32) {
                        char s [200],
                        *s = 0
                        wsprintf (s  "Cannot find file %s (or one of its
components)  Check to ensure the path and filename are correct and  
required libraries are available " cmd)
                        MessageBox(HWindow  s  "Application Execution Error"
MB_OK | MB_ICONEXCLAMATION)
        }
                else
                        cmdline->Clear ()
        }
        else
                DropDownList ()
}
```



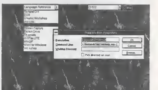
Promena *default* ponašanja List box-a



Dirke u kombinaciji sa ALT se hvataju pomoću WMSysKeyKown



Startovanje drugih programa iz naše Windows aplikacije



Startovanje drugih programa iz naše Windows aplikacije

KAPETANOV DNEVNIK

17.30

Razmišljao sam o koncepciji programa koji bi trebalo da zameni *Program Manager*. Buljeći u prazan ekran, pada mi na pamet da ga napravim na isti lazon kao *ParaWin*. Dakle, na ekranu će se nalaziti *combo box* sa WS_EX_TOPMOST atributom, tako da će uvek biti iscrtavan preko svih ostalih aplikacija. Kada se u edit kontrolu tog *combo boxa* otkuca neka komanda, sistem je izvršava. Recimo, dovoljno je otkucati pbrush i biće startovan *Paint Brush*. Otkucajte control i biće pozvan *Control Panel*.

Prava stvar, međutim, nastupa kada se pritiskom na strelicu nadole ili klikom miša spusti *list box* tog *combo box*-a. U njemu se nalazi spisak programa nalik na onaj u *Program Manager*-u, ali bez ikona. U taj spisak programa možemo dodavati pritiskom na Insert, menjati ih sa Alt-Enter, brisati ih sa Delete i naravno startovati sa Enter. Naravno, pri dodavanju programa sa Insert na ekranu se dobija *Program Item Properties* dijalog nalik na onaj iz *Program Manager*-a. U njega se unosi puno ime programa, njegov DOS path, radni direktorijum ili, ovo je novo, check box koji označava direktorijum unet pri startovanju programa. Kako sve ovo izgleda možete videti na slikama (1-4).

Varijanta dva za dizajn ovog programa jeste da se u *list box*-u pojave ikone, no to bi samo iskomplikovalo realizaciju, a pitanje je da li bi donelo kakvo poboljšanje. Zahtevalo bi više prostora na ekranu, a s obzirom da *Small Manager*, kako je ovaj nazvan, nema programske grupe, pitanje je da li bi sve što je potrebno moglo da stane.

17:40

Počinjem sa realizacijom *Small Manager*-a. Glavnom prozoru dajem stil WS_POPUP+WS_VISIBLE+WS_BORDER, što obezbeđuje da nema *titlebar* već samo veoma tanak okvir. Ovo me je svojevremeno dobro namučilo dok me neka dobra duša sa Sezama nije podučila kako da to uradim. Takođe, za *extended style* stavljam WS_EX_TOPMOST, što obezbeđuje da naš prozor stalno bude na vrhu. Isti taj stil sam pokušao pre jedno mesec dana da dodelim jednoj devojci, ali avaj, kompjuteri mogu isti zadatak da ponavljaju milionima puta, devojkama sve brzo dosadi. Nije ni čudo što se posle žele da više vremena provodimo sa kompjuterima nego sa njima.

Dimenzije prozora su 170×20 piksela, otprilike kao kod *ParaWin*-a. Kud svi Rusi, tud i mali Dejan.

17 45

Ubacujem *combo box* u glavni prozor i istog trenutka lupam glavom o zid: jedna od mogućnosti koju *ParaWin* ima jeste da ga možete uhvatiti mišem bilo gde na njegovoj površini osim na dugmetu i vući ga (*move*) po ekranu. Znači, nema *titlebar*-a već mu cela površina služi kao mesto pogodno za „hvatanje". Isti tip koji mi je na Sezamu otkrio dobitnu kombinaciju stilova odao mi je i tajnu kako se ovo radi. Jednostavno, na WM_LBUTTONDOWN poruku prozor odgovara sa jednim

SendMessage (HWindow, WM_SYSCOMMAND, SC_MOVE | HTCAPTION, 0l)

čime praktično samom sebi šalje poruku kojom „folira" da treba da bude pomeren. Posle toga *Windows* preuzima stvar i vi više nemate šta da brinete.

Na žalost, ova operacija se ne može vršiti na *combo box*-u. Njegova edit kontrola je fiksna i zbog toga se bez velike muke njena reagovanja na poruke ne mogu menjati. Padalo mi je na pamet da instaliram neki *Windows hook* za poruke ili da petljam sa *default* klasama prozora, ali to je sve previše komplikovano i ne isplati se. Žalo.

18:00

Posle deset minuta gledanja *combo box*-a sa *WinSpector*-om i ostalim alatkama za „provaljivanje" *Windows*-a, najzad se odlučujem da napravim svoj *combo box*. Pa nije to tako teško, majku mu, sve što treba jeste jedno dugme i jedna edit kontrola, čije ćemo ponašanje malo izmeniti tako da po potrebi „izbacuje" *list box*.

18 30

Uz jedano misteriozno zaglavljivanje *Windows*-a, program uspeva da proradi. Za sada je to samo prazna školjka koja izvršava komande iz edit kontrole i ništa više. Treba dodati odgovarajuću strukturu podataka za čuvanje informacija o programima, kao i *Program Item Properties* i *Choose Directory* dijaloge. Za podatke o programima izabrao sam jednostavan niz čiji članovi su strukture sa punim imenom programa, DOS nazivom, kao i „glavom" liste direktorijuma. Čuva se do deset direktorijuma koji su poslednji korišćeni kako bi korisnik pri startovanju programa mogao da izabere neki od njih.

18:55

Nezgodan „feature" pri definisanju dijaloga za izbor direktorijuma: napravio sam ga tako da bude veoma mali. Kasnije sam program *combo box* u kome su direktorijumi prikazani predstavio klasom TComboBox iz OWL-a i pri pozivanju njenog konstruktora definisao maksimalnu dužinu teksta. Na žalost, *Windows* se sa ovim nije slagao: pri unošenju teksta u *combo box* dužina je bila ograničena širinom *combo box*-a. A ja sam očekivao da će se tekst skrolovati po potrebi.

CONCURRENT C++

Ima već godina dana kako ETF u v deo da se zavrp e tac je programskog jezika Con C Na žalost morao sam da otputujem tako da nisam prisustvovao događaju Sudbina me je ipak sačekala pre nekih nedelju-dve upoznao sam autora samog jezika Dejana Milićeva On mi je dao nekoliko strana teksta iz kojih može da se vidi kako taj programski jezik radi Da budem iskren stvar deluje veoma impresivno

Razmislite Ima li šta logičnije nego da objekti koji treba da se preslikavaju iz realnog sveta mogu da rade neke poslove istovremeno? Pa na kraju krajeva, C++ i jeste nastao kao jezik za simulacije (nastao je na osnovama Simule i C-a) gde je paralelno izvršavanje neophodno To što smo mi na našim PC mašinama ograničeni na jednu programsku nit po programu to je minus naših operativnih sistema Ne mora da znači da treba da bude i minus naših programskih jezika.

Concurrent C++ je zasnovan na tri važna pojma

1. Nit (*thread*) predstavlja jedan proces u programu U okviru jednog programa može da se nalazi više niti koje bi se paralelno izvršavale Recimo jedna nit je zadužena za kontrolu miša i ona pomera onu strelicu na ekranu druga radi modemski I/O treća komunicira sa korisnikom i tako dalje Niti u svojoj suštini nemaju veze sa O-O konceptima – jedna nit je predstavljena najobičnijom funkcijom
2. Semafor (*semaphore*) predstavlja način komunikacije među različitim nitima Kako radi? Recimo da u programu imamo dve niti koje se paralelno izvršavaju Kada jedna nit želi da barata nekim podacima a druga u tome može da joj smeta onda ona postavi neki semafor (najobičnija klasa sa dve funkcije wait i signal) kojim signalizuje da su podaci u njenom vlasništvu Ukoliko sada druga nit želi da koristi podatke pa pokuša da postavi semafor ta nit se privremeno stopira sve dok ona prva ne oslobodi semafor Ovo je relativno slično otključavanju/zaključavanju slogova s tim što se cela stvar
   a) odvija samo u memoriji
   b) čim jedna nit pristupi semaforu sve ostale koje pokušaju da mu pristupe budu privremeno suspendovane dok se semafor ne oslobodi
3. Metaklase predstavljaju suštinu jezika Concurrent C++ Metaklasa se definiše veoma slično kao klasa onda se ona nadalje može koristiti umesto službene reči class Dozvolite da celu stvar ilustrujem primerom Recimo da želimo da napravimo ono što se u službenoj terminologiji naziva monitor, tj klasu unutar koja samo jedna funkcija može da se izvršava u isto vreme Dobar primer za takvu klasu jeste nekakav FIFO bafer sa dve funkcije get i put u kome ove dve funkcije ne smeju paralelno da se izvršavaju jer bi to izazvalo probleme Dakle treba da zabranimo da jedan proces pozove get dok drugi radi put i obratno

Prvo ćemo u običnom C maniru definisati klasu mux

```
class mux
    semaphore *sem
    mux (semaphore &sem)  sem (&e)   sem-wait ()
     mux ()   sem-signal ()
};
```

koja radi veoma jednostavno konstruktor postavlja neki semafor a destruktor ga oslobađa E sada definisati metaklasu monitor

```
metaclass monitor
    private    semaphore mutex
    ctor    mutex (1)
    pubin    mux dummy (mutex),
};
```

koja govori sledeće kada budemo definisali neku klasu na osnovu ove metaklase u privatni deo te klase će se ubaciti promenljiva **mutex** tipa **semaphore**. Ovo će biti semafor preko koga će se funkcije unutar klase „dogovarati"

Deklaracija **pubin** govori da u svaku *public* funkciju klasa definisanih na osnovu metaklase monitor treba kao ulazni kod dakle na samom početku funkcije dodati deklaraciju promenljive **dummy** tipa **mux** (klasu **mux** smo definisali malopre sećate se?) Ova deklaracija omogućuje da se odmah po ulasku u funkciju semafor postavi konstruktorom klase **mux** a da se oslobodi po izlasku iz funkcije destruktorom

I na kraju dolazi deklaracija klase **buffer** definisane na osnovu metaklase monitor

```
monitor buffer
    public
    void put (char c),
    char get  (),
};
```

kojom se postiže isključivost operacija u okviru ove klase

Naravno Concurrent C++ ne staje ovde moguće je zadati i statičke članove metaklasa kako bi se obezbedile ekskluzivnost pristupa pojedinačnim objektima kompajler je dovoljno pametan da prepozna funkcije-inspektore (one sa **const** deklaracijom) tako da se može napraviti i slučaj u kome je njima dopušten istovremeni pristup dok mutatorima nije i slično

Malo sam se ugnzao za jezik dok sam kucao ono kompajler Radi se više o pretprocesoru čija je funkcija da Concurrent C++ prevede u C++ Ovakav pristup pomalo mrzak nama „modernim" programerima ušuškanim u integrisane okoline je sasvim prikladan CC++ je još uvek u povoju jednostavno se ne isplati razvijati ga kao pravi kompajler Moja razmišljanja idu čak dotle da je i pretprocesor suvišan i da bi možda CC++ trebalo realizovati isključivo u okvirima C++ a no pitanje je da li je to uopšte moguće Kao prvo bilo bi veoma teško realizovati ekvivalent **pubin** deklaracije (koja ubacuje neki kod na početak svake funkcije) na golom C++-u mada ne i nemoguće – izuzeci su rešenje i za tu boljku

Čini mi se da mi je Strustrup jednom prilikom rekao da koristi neku HP paralelnu mašinu tako da ću ga prvom prilikom pitati šta misli o CC++ u Ko zna možda i on planira neke konkurentne ekstenzije za C++? Tema je u svakom slučaju veoma zanimljiva tako da ćemo se definitivno vraćati na nju i u budućnosti

19 20

Program radi i to dobro Treba samo još dodati učitavanje i snimanje podataka na disk i stvar će biti završena S obzirom da je struktura u kojoj se čuvaju podaci o programima relativno jednostavna a da mi je u isto vreme dužina programa bitna stavka I/O ne radim pomoću C++ tokove (*streams*) već pomoću *Windows* funkcija **_lopen, _lread, _lwrite, _lclose** i **_lcreat** Rad sa njima je identičan radu sa fajl handlovima, tako da to ne predstavlja problem

19 40

Program je oficijelno završen! Sada ga samo treba podrobno istestirati i može da ide u svet

SHAREWARE POD SANKCIJAMA

Small Manager sam proglasio za *shareware* Cena mu je pet maraka za pojedince i sto maraka za organizacije Da budem iskren nemam pojma da li će mi iko poslati pare Domaće *shareware* tržište praktično ne postoji a o stranom zbog sankcija ne vredi ni razmišljati No videćemo U svakom slučaju ću vas obavestiti šta je bilo

Pošto sam ipak zainteresovan da *Small Manager* izađe u svet makar kao reklama razmišljao sam kako da to realizujem Bilo bi glupo da za ljude u Jugi bude *shareware*, a za ostale *freeware* Padalo mi je na pamet da *Small Manger* proglasim za *Yugoware* Da biste ga koristili morate da pročitate sledeći tekst

I am a Serb Until a few years I used to watch what the Iraqi were doing and think that they are a primitive, bloodthirsty nation Now everybody is looking at us Serbs that way But please consider this

Only 30% percent of our population voted for the political party that brought this war on all of us Unfortunately by clever use of the election system and by means of force (the police is theirs) they are still ruling this country

So next time you think of Serbs don't think of us as killers Most of us are normal people just like me and you

U vreme pisanja ovog teksta još uvek nisam siguran da li da ubacim ovu *Yugoware* opciju za strance čini mi se da bih samo zakomplikovala stvar Najverovatnije će *Small Manager* ipak ostati samo *shareware* pa makar se ti stranci vozali za džabe Da nam se malo osvete što Tijanić krade njihove filmove

*) Autor je nezavisni konsultant. Možete ga kontaktirati na Partizanske 70 11090 Beograd ili na djelovic@ezsezam yu

NEURONSKE MREZE I BICEPSI

Do pre dva godine imao sam običaj da svakodnevno plivam po dve sata Koliko je to dosadan sport nemam reči da vam opišem Dok obrćete krugove razmišljate o kojećemu a posle počnete da brojite pločice na dnu bazena

Kada sam pre tri meseca krenuo u teretanu shvatio sam da je tek to dosadno Tamo – za ime boga! – nema čovek s kim da popriča Uobičajena populacija teretana su momci sa par kila zlata oko vrata ćelavi sa čelom od dva prsta Oni su dosadniji i od plivanja

Anyway u silnoj dokolici prilikom vežbanja čovek razmišlja o svemu pa mi je tako palo na pamet da napravim neuronsku mrežu koja bi mi govorila koje vežbe da radim koliko kojim redosledom i koliko pauze da pravim između njih Počeo sam sa sakupljanjem podataka merio sam redovno sve živa obime na telu i zapisivao tačno šta sam od vežbi radio Kada bih stigao kući unosio sam podatke u kompjuter i pomalo doterivao mrežu Zadatak mreze minimizirati razliku između idealnog i trenutnog obima svakog dela tela.

Da skratim priču od pre dve nedelje prestao sam da slušam bezvezne u teretani i počeo sam da slušam kompjuter Posle svakog odlaska u teretanu dodajem nove podatke tako da mreža ima *feedback* o tome kakve rezultate daje Još uvek je rano da bi se videlo da li ovaj način rada daje dobre rezultate ali za sada sve ide kako treba Loše je jedino to što je mreža gotovo u potpunosti izbacila sve vežbe za kičmu – one služe samo da je ojačaju ali ne i da je povećaju tako da ih mreža smatra za nepotrebne Kako ovo izbeći nemam pojma Jedan moj drugar inače zaljubljenik u mreže ubeđuje me da će mreža to da proveli sama Ako i to uradi – zameniću mozak kompjuterom.

IZ MENIJA U MENI

Meniji su elegantan i jednostavan način organizacije složenog programa. Korisnik može lagodno da se šeta kroz opcije i jednim pritiskom tastera da ulazi u programe koji se skrivaju iza naziva pojedinih stavki. Izuzetno lepo za upotrebu i – izuzetno dosadno za programiranje. Na žalost, većinu kreativne energije usmeravamo ka ključnim, izvršnim delovima programa, dok interfejs ka korisniku redovno zapostavljamo. Sve bi bilo drugačije kada bismo imali na raspolaganju jednu malu biblioteku funkcija za rad sa menijima...

Jovan Skuljan

Kada pišemo neki složeniji program koji u sebi sadrži veći broj različitih opcija, postavlja se pitanje kako organizovati sve elemente u preglednu celinu. Razgranato stablo menija i podmenija nameće se tu da kao izuzetno efektno, a relativno jednostavno rešenje.

Pod jednim menijem podrazumevaćemo spisak opcija na ekranu, pri čemu korisnik treba da izabere jednu od njih. Izbore u stilu „Ukucajte redni broj 1-17 i pritisnite ENTER" odbacićemo kao suviše školjinske za naše mogućnosti, dok bi *Windows* ikonice i slična grafička egzibicija mogle još malo da popričekaju. Opredelićemo se za meni kroz koji se može šetati strelicom gore-dole, pri čemu se osvetljava jedna tekuća stavka. Kada osvetlimo stavku koja nas zanima, pritiskom na ENTER pokrećemo željeni program. Po povratku iz programa opet se nalazimo u istom meniju, sa sačuvanom tekućom stavkom koju smo zadnju osvetlili. Ako se neka od stavki i sama dalje grana, ulaskom u nju dobićemo novi meni, sa novim spiskom stavki kroz koji se možemo kretati kao i kroz prethodni. Tako možemo pročešljati celo stablo opcija i steći sliku o tome kako su pojedini programi organizovani. Pri svakom izlasku iz menija pamti se tekuća osvetljena stavka i obnavlja se pri povratku u taj meni.

Toliko što se tiče ideje. Da kažemo reč-dve i o realizaciji. Svaki C programer je već nakon prvih iskustava imao priliku da primeti kako standardan C sa svojom **printf** funkcijom i nije baš naročito podesan za nekakve egzibicije po ekranu: niti je moguće postaviti kursor na željeno polje ekrana, niti se može birati atribut za ispis teksta. Umesto toga, ispis može mo vršiti red za redom, uz neizbežno skrolovanje, kao u ona čika vremena kada smo koristili stari dobri FORTRAN (daleko bio i ne ponovio se). Noviji kompajleri, doduše, donose i funkcije za rad sa ekranom, ali to je sve daleko od standarda.

Nije nikakav principijelan problem napisati funkcije za rad sa ekranom – svaki PC u svom BIOS-u ima gotove potprograme za to. Međutim, kako ih pozvati? Rešenje koje uvek postoji jeste da se takve funkcije razviju u asembleru, a da se kasnije priključe našem programu pomoću linkera. Izvodljivo, ali ružno. Zar da potežemo asembler za samo par mašinskih instrukcija?

Srećom, Microsoft-ov C ima ugrađen asembler, što nam omogućava da elegantno kombinujemo dva jezika. Naravno, i to je van standarda, ali je bar sve pod našom kontrolom.

RAZGRANATO STABLO

Dakle, pošto smo rešili problem rada sa ekranom, možemo se ponovo vratiti na organizaciju menija. Prosto radi primera, napisaćemo program čije je stablo opcija dato na slici 1. Nije uopšte važno šta konkretno znače pojedine opcije, niti kakvi se misteriozni programi za obradu iza njih kriju. Važno je samo kako su opcije povezane i kako se iz jedne može doći u drugu.

Osnovni meni ima tri stavke (slika 2), od kojih se prva dalje grana na novi podmeni od četiri stavke, dok druga ulazi u podmeni sa tri stavke. Od te tri stavke druga se grana na dve nove. Ukupan broj menija iznosi četiri. Svaki od njih ima svoj naslov, svoj spisak stavki i svoj spisak funkcija u koje se ulazi pri izboru svake stavke. Dakle, svaki meni bi mogao da ima neku svoju *strukturu podataka* koja sadrži kompletnu informaciju o njemu. Sam program za rad sa menijima mogao bi biti univerzalan, a kao ulazni argument primao bi samo adresu strukture koja nosi informaciju o konkretnom meniju.

Program ćemo razvijati u modulima. To znači da ćemo pojedine funkcije grupisati u zasebne izvorne datoteke koje ćemo zasebno kompajlirati, a kasnije linkovati u konačni izvršni program. Prednost ovakvog načina rada nad manirom da se sva trpa u istu datoteku je višestruka, čak i kada količina napisanog koda nije baš tako velika da bi trebalo vršiti razbijanje. Pre svega, jednom napisane i proverene funkcije ne moraju se više kompajlirati zato što menjamo neke druge delove programa. Osim toga, u različitim modulima možemo koristiti ista imena globalnih promenljivih, stavljajući ispred njih deklaraciju **static**. To je zgodno ako imamo veći broj veoma sličnih modula, kao što su kod nas moduli za pojedine menije, gde je najjednostavnije koristiti iste nazive promenljivih.

Jasno, mogla bi se staviti i značajna zamerka na modularni rad, u smislu da povećani broj datoteka otežava vođenje evidencije, ali, kao što ćemo kasnije videti, posao oko evidencije i povezivanja modula prepustićemo programu NMAKE, koji se upravo za te svrhe isporučuje uz svaki kompajler.

Mi smo program razbili na ukupno osam modula, odnosno osam izvornih datoteka, organizovanih na sledeći način:

IZBOR.C – Glavni program (main)
CONSOLE.C – Funkcije za rad sa ekranom i tastaturom
COMMON.C – Neke zajedničke funkcije za sve module
CHOICE.C – Potprogrami za rad sa menijima
OSNOVNI.C, DATOTEKE.C, PREGLED.C i GRAFIKA.C – Pojedini meniji

Ovom spisku treba dodati i posebnu datoteku *datadef.h* koja sadrži sve definicije i deklaracije zajedničke za sve module.

NOVI TIPOVI PODATAKA

Prvi deo datoteke *datadef.h* sadrži definicije struktura i drugih tipova podataka koji se koriste u programu. Uvođenjem novih tipova, a naročito struktura, znatno se povećava preglednost programa i olakšava njegovo pisanje. Mi smo uveli tri strukture (KOORDINATE, IZBOR_INFO i TEKST) i jedan pointer na funkciju (PROCEDURA). Možda ovaj poslednji tip deluje najmisterioznije, pa ćemo pokušati da ga malo protumačimo.

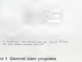
Pointeri u C-u su izuzetno moćno oruđe, bez koga se ne može zamisliti neki bolji program. Navikli smo uglavnom na promenljive koje sadrže neke uobičajene podatke, kao što su celi brojevi (*int*), racionalni brojevi (*float*, *double*) ili karakteri (*char*). Takođe se moramo navići i na promenljive koje sadrže *adrese* različitih objekata. Na primer, ako deklarišemo promenljivu **ceo_broj** kao **int** *ceo_broj, onda ta promenljiva može da sadrži adresu nekog celog broja ili čak niza celih brojeva, kojima ćemo pristupati sa **ceo_broj[0]**, **ceo_broj[1]** i tako dalje. Slično tome, ako neku promenljivu **celobrojna_funkcija** deklarišemo sa **int** (*celobrojna_funkcija)(char), dobijamo promenljivu koja može sadržati adresu neke funkcije tipa *int* sa jednim argumentom tipa *char*. U programu kasnije možemo pozivati takvu funkciju prosto sa **k** = celobrojna_funkcija('a') ili nekako slično. Podrazumeva se (a to isto važi za sve pointere u C-u) da prethodno moramo promenljivoj dodeliti vrednost, recimo sa **celobrojna_funkcija** = malo_slovo, gde je **int malo_slovo(char)** neka stvarna funkcija, definisana u našem programu, koja zaista nešto radi (recimo, proverava da li je zadati karakter malo slovo i vraća jedinicu ako jeste, a nulu ako nije).

Ako ispred deklaracije promenljive stavimo **typedef**, onda umesto deklarisane promenljive imamo uveden tip podataka koji kasnije možemo ravnopravno koristiti sa ostalim tipovima u deklaracijama promenljivih. Naš tip PROCEDURA predstavlja adresu funkcije koja nema argumenata i koja ne vraća nikakav rezultat. Ako želimo da uvedemo stvarnu promenljivu tog tipa, kucaćemo nešto poput **PROCEDURA funkcija**.

Sada se možemo vratiti na strukture definisane u datoteci *datadef.h*. Najjednostavnija struktura je KOORDINATE, koja sadrži red i kolonu za ispis na ekranu (ekran ima 25 redova i 80 kolona u standardnom tekstualnom režimu). Struktura TEKST opisuje tekst koji će se štampati na ekranu. Zadaju se pozicija (red i kolona), atribut i sadržaj teksta. Konačno, struktura IZBOR_INFO daje kompletnu informaciju o izbornom meniju koji će se pojaviti na ekranu. Nazivi elemenata strukture, koje smo koristili u deklaraciji, su jasni sami za sebe. Pre svega, tu je ukupan broj stavki u meniju, zatim dužina stavki (odnosno širina menija), a predvideli smo i poseban element za čuvanje rednog broja tekuće stavke menija. U početku je to nula (stavka u samom vrhu menija), ali kako se budemo šetali kursorom gore-dole, prelazićemo na stavke sa različitim rednim brojem. Pošto se taj redni broj uvek pamti, kursor će ostati na zadnjoj izabranoj stavci čak i kada napustimo meni, pa se kasnije ponovo u njega vratimo.

Preostali elementi strukture IZBOR_INFO sadrže naslov menija (koji će se pojaviti u zaglavlju ekrana) i, najzad, niz naziva stavki, kao i niz adresa funkcija (poslova) koje odgovaraju svakoj stavci, odnosno koje će se izvršavati kada pritisnemo ENTER na izabranoj stavci.

Drugi deo datoteke *datadef.h* sadrži deklaracije svih funkcija koje imaju šanse da budu pozivane iz različitih modula.



Slika 1. Osnovni izbor programa

MAŠINSKO PROGRAMIRANJE U C-u

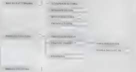
Microsoft je korisnicima svog C kompajlera ponudio izuzetno elegantan način da u program ugrade mašinske blokove pisane standardnim asemblerskim instrukcijama. Drugim rečima, kompajler istovremeno može da radi i kao asembler, što u drugim jezicima do sada nije viđeno. Prednosti ovakve koncepcije ne mogu se ni nabrojati, a kamoli ilustrovati, samo u par redova, koliko mi nameravamo ovom prilikom o tome da govorimo. Tema zaslužuje barem poseban članak, ali ćemo pokušati da i ovde donekle razjasnimo

Listing 1

```
#                                   MAKEFILE
#
izbor exe   izbor obj b1 lib b2 lib
   tlink /3d izbor, NUL b1 b2
b1 lib   choice obj common obj console obj
   if exist b1 lib del b1 lib
   lib b1+choice+common+console
b2 lib   osnovni obj datoteke obj pregled obj grafika obj
   if exist b2 lib del b2 lib
   lib b2+osnovni+datoteke+pregled+grafika,
izbor obj   izbor c
   cl /AL /Od /Zi /c izbor c
choice obj   choice c
   cl /AL /Od /Zi /c choice c
common obj   common c
   cl /AL /Od /Zi /c common c
console obj   console c
   cl /AL /Od /Zi /c console c
osnovni obj   osnovni c
   cl /AL /Od /Zi /c osnovni c
datoteke obj   datoteke c
   cl /AL /Od /Zi /c datoteke c
pregled obj   pregled c
   cl /AL /Od /Zi /c pregled c
grafika obj   grafika c
   cl /AL /Od /Zi /c grafika c
/*                                  DATADEF H
*/
typedef void (*PROCEDURA)(void)
typedef struct
{
   int red kolona
}
KOORDINATE
typedef struct
{
   int ukupno duzina stavka
   char *naslov **naziv
   PROCEDURA *posao
}
IZBOR INFO
typedef struct
{
   int red kolona,atribut
   char *sadrzaj
}
TEXT
void set_cursor_position(int row int col)
void set_cursor_size(int first int last)
void repeat_char(char chr int attrib int num)
void print char(char chr int attrib int num,int row int col)
void clear screen(void)
void print extra(char *text int len,int attrib,int row int col),
void centriraj(char *text int row,int attrib)
void write_text(TEXT text)
void linija(char *niz int attrib int row,int col int len)
int stat_tester(void)
void naglavlje(void)
void prazan(void)
void obrada(IZBOR INFO *izbor)
void osnovni(void),
void datoteke(void)
void pregled(void)
void grafika(void)
/*                                  IZBOR C
*/
#include  datadef h
main()
{
   set_cursor_size(32,0)
   clear_screen()
   osnovni()
   clear_screen()
   set_cursor_size(12 13);
}
/*                                  CONSOLE C
*/
#include <bios h>
#include <string h>
#include  datadef h"
void set cursor_position(int row int col)
{
   _asm
   {
                 mov   ah 2
                 xor   bh,bh
                 mov   dh row
                 mov   dl col
                 int   10h
   }
}
void set_cursor_size(int first int last)
{
   asm
   {
                 mov   ah,1
                 mov   ch,first
                 mov   cl,last
                 int   10h
   }
}
void repeat_char(char chr int attrib,int num)
{
   _asm
   {
                 mov   ah 9
                 mov   al,chr
                 mov   bl,attrib
                 xor   bh,bh
                 mov   cx num
                 int   10h
   }
}
void print char(char chr,int attrib int num,int row int col
{
   set_cursor_position(row,col)
   repeat_char(chr,attrib num)
}
void clear screen(void)
{
   print_char(   ,0x07 2000 0 0)
}
void print extra(char *text int len int attrib int row int col)
{
   int i
   for (i=0 text[i]!=0 i++) print_char(text[i] attrib 1 row col i)
   for (i i<len i++) print char(    attrib 1 row col+i)
}
void centriraj(char *text,int row int attrib)
{
   int len,i,
   len=strlen(text)
   for (i=0,text[i]!=0,i++) print_char(text[i] attrib 1 row (80 len)/2 i)
}
void write_text(TEXT text)
{
   int i,
   for (i=0,text sadrzaj[i]!=0 i++)
      print_char(text sadrzaj[i] text atribut 1 text red,text kolona+i)
}
void linija(char *niz int attrib int row,int col int len)
{
   print char(niz[0] attrib 1 row col),
   print_char(niz[1] attrib,len 2 row col+1)
   print_char(niz[2] attrib,1 row col+len 1)
}
int stat_tester(void)
{
   int i
   i=getch()
   if (i != 0)
      return(i)
   else
      return(1000+getch())
}
/*                                  COMMON C
*/
#include  datadef h
void zaglavlje(void)
{
   linija("\xC9\xCD\xBB" 0x0F, 0  0 80)
   linija("\xBA\x20\xBA",0x0F  1  0 80)
   linija("\xBA\x20\xBA  0x0F  2, 0,80)
   linija( \xC8\xCD\xBC  0x0F  3  0,80)
   linija("\xCD\xCD\xCD  0x0F,20, 0,80),
   centriraj( Demonstracija rada sa menijima' 1 0x0F)
}
void prazan(void)
{
   return,
}
/*                                  CHOICE C
*/
#include  datadef h"
#define AKTIVNO  0x70
#define PASIVNO  0x07
TEXT uputstvo[2]=
 { {23,1,0x0F "Strelicama gore i dole izaberite program i pritisnite ENTER  }
   {24,1,0x0F "Ako zelite izlazak, pritisnite ESCAPE  } }
KOORDINATE koordinate stavke(IZBOR_INFO *izbor)
{
   KOORDINATE pozicija,
   pozicija red = 13   izbor >ukupno + 4 izbor >stavka
   pozicija kolona = (80 izbor >duzina) / 2
   return(pozicija),
}
void stampaj stavku(IZBOR_INFO *izbor int atribut)
{
   KOORDINATE pozicija,
   pozicija=koordinate_stavke(izbor)
   print extra(izbor >naziv[izbor >stavka]
               izbor >duzina atribut,pozicija red pozicija kolona)
}
void formiraj_izbor(IZBOR INFO *izbor)
{
   int i broj_stavke
   KOORDINATE polje
   clear_screen()
   zaglavlje()
   centriraj(izbor >naslov 2 0x0F)
   broj_stavke=izbor >stavka
   for (izbor >stavka=0 izbor >stavka<izbor >ukupno izbor >stavka++)
   {
      polje=koordinate stavke(izbor),
      if (izbor >stavka == 0)
         linija("\xC9\xCD\xBB  0x07 polje red 1 polje kolona 2 izbor
>duzina+4)
      else
         linija("\xCC\xCD\xB9  0x07 polje red 1 polje kolona 2 izbor
>duzina+4)
      linija("\xBA\x20\xBA" 0x07,polje red,polje kolona 2 izbor >duzina+4)
      stampaj_stavku(izbor,PASIVNO)
   }
   linija("\xC8\xCD\xBC  0x07 polje red+1,polje kolona 2 izbor >duzina+4)
   izbor >stavka=broj_stavke
   for (i=0 i<2 i++) write text(uputstvo[i])
}
void obrada(IZBOR_INFO *izbor)
{
   int kraj potrebno tester
```

```
kraj=0,potrebno=1
do
{
    if (potrebno) formiraj_izbor(izbor) potrebno=0
    stampaj_stavku(izbor AKTIVNO)
    taster=uzmi_taster()
    stampaj_stavku(izbor PASIVNO),
    switch(taster)
    {
        case 1072  if (izbor->stavka > 0) izbor->stavka--
                   break
        case 1080  if (izbor->stavka+1 < izbor->ukupno) izbor->stavka++
                   break
        case 13    izbor->posao[izbor->stavka]();
                   potrebno=1
                   break
        case 27    kraj=1
                   break
    }
}
while(!kraj);
}
/*                              OSNOVNI C
*/
#include "datadef h
static char naslov[]="OSNOVNI IZBOR PROGRAMA"
static char *naziv[]={ RAD SA DATOTEKAMA
                      "PREGLED PODATAKA
                      OBRADA PODATAKA  }
static PROCEDURA posao[]= { datoteke
                           pregled
                           prazno }
static IZBOR INFO izbor={ 3  38  0, naslov  naziv  posao }
void osnovni(void);
{
   choice(&izbor)
}
/*                              DATOTEKE C
*/
#include  datadef h
static char naslov[]= RAD SA DATOTEKAMA"
static char *naziv[]={ "UCITAVANJE SA DISKA"
                       SNIMANJE NA DISK
                      "BRISANJE SA DISKA
                      "PROMENA NAZIVA  }
static PROCEDURA posao[]= { prazno
                           prazno,
                           prazno
                           prazno }
static IZBOR INFO izbor={ 4  38  0  naslov  naziv  posao }
void datoteke(void)
{
   choice(&izbor)
}
/*                              PREGLED C
*/
#include  datadef h
```

```
static char naslov[]="PREGLED PODATAKA"
static char *naziv[]={ "TEKSTUALNI PRIKAZ"
                      "GRAFICKI PRIKAZ
                      STAMPANJE  }
static PROCEDURA posao[]= { prazno
                           grafika,
                           prazno }
static IZBOR_INFO izbor={ 3  38  0, naslov, naziv  posao },
void pregled(void)
{
   choice(&izbor)
}
/*                              GRAFIKA C
*/
#include  datadef h
static char naslov[]= GRAFICKI PRIKAZ
static char *naziv[]={ "NISKA REZOLUCIJA ,
                      VISOKA REZOLUCIJA" }
static PROCEDURA posao[]= { prazno
                           prazno }
static IZBOR_INFO izbor={ 2  38  0  naslov  naziv  posao },
void grafika(void)
{
   choice(&izbor),
}
```



Slika 2. Stablo opcija

osnovna principe takvog „dvojezičnog" programiranja.

Zašto bi uopšte neko osećao potrebu za mašinskim jezikom unutar C-a? Zvuči gotovo kao padanje sa konja na magarca kada se ima u vidu samo efikasnost izvornog koda: jedna jedina C naredba može da radi veliki posao, dok jedna mašinska naredba ne radi praktično ništa. Pa ipak, potreba za mašinskim blokom može se javiti, i to barem u dva slučaja:

a) Neki posao se u C-u jednostavno ne može uraditi (navodno, ali istinito).
b) Neki posao u C-u se neprihvatljivo sporo odvija. Postoji možda i treći slučaj, ali on, izgleda, ne može ući u objektivno razmatranje:
c) Vuče vas nostalgija za mašincem još iz ranih dana.

Neko će možda reći da ni slučaj pod b) ne treba shvatiti ozbiljno jer je čisto izvoljevanje. Bilo kako bilo, svako će primeniti neki svoj kriterijum i dozirati mašinske delove u onoj meri u kojoj on smatra da je to neophodno. Sigurno je jedino to da takvih delova ne treba da bude mnogo.

Što se same tehnike tiče, ona je vrlo prosta. Bilo gde u svom programu možemo ukucati direktivu _asm a zatim u izdvojenom bloku { } ukucati niz mašinskih naredbi kao u asembleru (uključujući čak labele skokova i potprograme). Moramo voditi računa da na kraju svega toga registri DS, SI, DI i BP moraju ostati nepromenjeni, a ako smo koristili STD (setovanje indikatora smera), moramo obavezno resetovati indikator smera sa CLD.

Ukoliko se mašinski blok javi na kraju neke funkcije u C-u, onda sadržaj registara DX i AX daju neposredno izlaznu vrednost funkcije, zavisno od tipa (samo AX za cele brojeve do 16 bita, a DX:AX za 32-bitne podatke ili adrese).

Mi smo koristili mašinske blokove samo za pozivanje standardnih funkcija BIOS-a (postavljanje položaja i veličine kursora, kao i ispis karaktera na ekranu sa zadatim atributom).

OPIS FUNKCIJA

Verujemo da su listinzi programa koje objavljujemo dovoljni za analizu, pa ćemo samo kratko dati opise pojedinačnih funkcija.

main (glavni program) sastoji se samo od nekoliko elemenata. Kursor se uklanja, ekran se briše i poziva se osnovni meni. Nakon izlaska iz menija ponovo se briše ekran, obnavlja se kursor i vrši se povratak u DOS.

set_cursor_position postavlja kursor na zadati red (*row*) i kolonu (*col*) ekrana. Redovi su numerisani od 0 do 24, a kolone od 0 do 79.

set_cursor_size određuje veličinu kursora. Zadaju se prva (*first*) i poslednja (*last*) linija u rasponu od 0 do 13, što pokriva celo polje za ispis karaktera. Standardni kursor u DOS-u pokriva samo najniže dve linije karaktera (*first*=12, *last*=13). Kursor se može učiniti nevidljivim stavljajući *first*=32.

repeat_char ispisuje na ekranu (na trenutnoj poziciji kursora) zadati karakter *chr* sa zadatim atributom *attrib*, pri čemu se karakter ponavlja *num* puta. Kursor se pri tome ne pomera, već ga mi moramo postaviti za ispis sledećeg karaktera. Atribut je osmobitni broj u koma svaki bit nosi neku informaciju prema sledećoj šemi:

Bit 7 – Treptanje karaktera (blink)
Bit 6 – Crvena komponenta pozadine
Bit 5 – Zelena komponenta pozadine
Bit 4 – Plava komponenta pozadine
Bit 3 – Intenzitet (sjaj) karaktera
Bit 2 – Crvena komponenta karaktera
Bit 1 – Zelena komponenta karaktera
Bit 0 – Plava komponenta karaktera

Ako zanemarimo sjaj i treptanje, najjednostavnije je posmatrati atribut kao dvocifren heksadekadni broj, pri čemu prva (viša) heksadekadna cifra daje boju pozadine, a druga (niža) cifra daje boju samog karaktera. Tako, na primer, atribut 0x07 predstavlja standardna bela slova na crnoj podlozi, dok 0x70 predstavlja invertovane karaktere (crna slova na beloj podlozi). Karakteri pojačanog sjaja (opet bela slova na crnoj podlozi) bili bi opisani atributom 0x0F. Ukoliko imate monohromatski ekran, tu bi se negde i iscrple sve mogućnosti. Kolor grafičke kartice dopuštaju, naravno, sve kombinacije boja. Međutim, podosta eksperimentisanja je potrebno da bi se postigao efekat prijatan za gledanje.

print_char ispisuje zadati karakter *chr* sa zadatim atributom *attrib* i brojem ponavljanja *num* u zadatom redu *row* i koloni *col*.

clear_screen briše ekran i postavlja kursor u gornji levi ugao.

print_extra ispisuje zadati tekst *text* u polju ukupne dužine *len* sa atributom *attrib*, sa početkom u zadatom redu *row* i koloni *col*. Podrazumeva se da je tekst kraći od zadate dužine polja. Dopuna do pune dužine vrši se blankovima sa zadatim atributom.

centriraj ispisuje zadati tekst *text* centrirano u zadatom redu *row*, sa atributom *attrib*.

write_text ispisuje zadati tekst opisan strukturom *text*. Struktura sadrži koordinate i atribut za ispis.

linija ispisuje na ekranu liniju, odnosno niz karaktera definisanih zadatim *uzorkom* i atributom *attrib*, sa početkom u zadatom redu *row* i koloni *col* i sa ukupnom dužinom *len*. Uzorak je niz od tri karaktera koji definišu izgled linije, pri čemu se srednji karakter ponavlja onoliko puta dok se ne postigne zadata dužina.

uzmi_taster čeka da korisnik ukuca neki taster i vraća odgovarajući kod. U slučaju standardnih ASCII karaktera (kodovi do 255) vraća se sam ASCII kod, dok se za specijalne tastere vraća njihov specijalni kod uvećan za 1000. Tako, na primer, funkcijski taster F1 dobija vrednost 1059, strelica gore 1072, taster Home 1071 i tako dalje. Ukoliko nemate pri ruci kom

pletnu tabelu kodova, najbolje da je sami formirate probom pomoću ovog potprograma.

zaglavlje formira osnovni izgled ekrana sa zaglavljem u vrhu i jednom horizontalnom linijom pri dnu. Sredina ekrana ostaje prazna za menija, a donji deo ekrana je rezervisan za poruke korisniku.

prazno ne radi ništa. Koristi se za one grane menija koje još nemaju nikakvu funkciju.

koordinate stavke vraća koordinate tekuće stavke u zadatom meniju *izbor*.

štampaj stavku obavlja štampanje tekuće stavke u zadatom meniju *izbor* sa zadatim *atributom*. Radi preglednosti, mi smo definisali dva atributa za štampanje stavke: AKTIVNO (0x70), kada je stavka trenutno izabrana, i PASIVNO (0x07), kada stavka nije izabrana.

formiraj izbor ispisuje zaglavlje i zadati meni *izbor*, pri čemu se slika centrira u odnosu na granice ekrana. Najpre se iscrtava *zaglavlje*, ispisuje naslov menija, a zatim se postavlja mreža (okvir) sa ispisanim nazivima stavki. Konačno, u dnu ekrana se pojavljuje uputstvo za rad sa menijem.

choice je ključni program u radu sa menijima. On obavlja kompletan rad sa zadatim menijem *izbor*: ispisivanje menija na ekranu, čekanje da korisnik ukuca neki taster, šetanje po meniju gore-dole, izbor zadate stavke i ulazak u odgovarajući program, sve dok se ne ukuca *Escape* za izlazak. Dve celobrojne promenljive se čudnim nazivima, *potrebno* i *kraj* služe za kontrolu rada programa. Ako je *potrebno* = 1, to je signal da treba obnoviti meni na ekranu (obavezno posle povratka iz neke izabrane opcije, kao i na samom početku rada sa menijem). Lakonska konstrukcija *if (potrebno)* (ako je potrebno) u stvari je skraćenica za *if (potrebno == 1)*. Promenljiva *kraj* postavlja se na jedinicu samo ako korisnik ukuca *Escape*, i to je onda signal za kraj rada sa menijem, odnosno povratak iz potprograma. Ciklus je opet zapisan lakonski: *do ... while(!kraj)* (radi ... dok nije kraj), jer je to mnogo jasnije od *while(kraj == 0)* ili *while(kraj != 1)*. U ciklusu (osim ispisivanja menija po potrebi) nema mnogo operacija. Tekuća (aktivna) stavka se „osvetljava" čeka se ukucani taster, stavka se „gasi" i ulazi se u grananje na osnovu ukucanog tastera. Strelice gore i dole pomeraju tekuću stavku u zadatom smeru (ukoliko je to moguće), dok taster ENTER poziva funkciju koja je dodeljena tekućoj stavci.

osnovni, **datoteka**, **pregled** i **grafika** se prosto svode na poziv programa *choice* za svaki od zadatih menija. Osnovni posao oko svakog menija je priprema strukture *izbor*.

NMAKE I MAKEFILE

Pri modularnom razvoju većih programa vrlo bi naporno bilo da sami vodimo računa o svim izmenama koje vršimo na izvornim datotekama. Da li smo kompajlirali neki modul nakon editovanja? Da li smo linkovali izmenjen modul sa ostalim objektnim datotekama? Na ova i slična naprijatna pitanja odgovor je vrlo kratak: NMAKE.

Program NMAKE (skraćeno od New MAKE) poziva se vrlo jednostavno, bez ikakvih argumenata, ukoliko sve naše želje u vezi sa kompajliranjem i linkovanjem našeg programa upišemo u specijalnu datoteku zvanu MAKEFILE (bez nastavka). Čitajući tu datoteku, program će pronaći informacije o tome koje izvorne datoteke ulaze u projekat, kako se iz njih dobijaju objektne datoteke, a kako konačna izvršna datoteka. Pošto sve ova datoteka na disku pod DOS-om imaju upisan datum i vreme poslednje izmene, program je u stanju da obavi samo one neophodne poslove nad datotekama koje su menjene u međuvremenu (od poslednjeg poziva NMAKE). To je velika prednost programa NMAKE u odnosu na neku BAT datoteku slične namene, koja bi uvek izvršavala isti niz komandi bez obzira da li smo menjali samo jednu ili sve izvorne datoteke.

Očigledno, ključ primene programa NMAKE je već pomenuta datoteka MAKEFILE. Njen sadržaj može biti prilično komplikovan, ali ćemo mi pokušati da damo recept za najjednostavniju moguću verziju koja još uvek uspešno obavlja posao.

Datoteka MAKEFILE sastoji se od jednog ili više izdvojenih blokova (ukupno 11 u našem primeru), pri čemu svaki blok ima praktično istu strukturu: prva linija bloka kaže *šta se pravi* (koja datoteka se formira) i *od čega ona zavisi*, dok ostali redovi opisuju *kako se to pravi*, odnosno koje komande treba izvršiti da bi se dobila zadata datoteka. Uzmimo kao primer prvi blok naše MAKEFILE datoteke. Blok počinje linijom

```
izbor.exe: izbor.obj b1.lib b2.lib
```

što znači da treba formirati datoteku *izbor.exe*, u čijoj izgradnji učestvuju datoteke *izbor.obj*, *b1.lib* i *b2.lib*. Primetimo da ova linija (a to važi za sva početna linije u bloku) počinje od same leve ivice reda (bez uvlačenja). To je signal programu NMAKE da se radi o novom bloku.

Drugi red u našem primeru glasi:

```
    link /CO izbor, NUL b1 + b2
```

što prosto predstavlja poziv linkera da bi se objektna datoteka *izbor.obj* povezala sa bibliotekama *b1.lib* i *b2.lib* u konačni izvršni program *izbor.exe*. Opcija /CO nije obavezna, ali je poželjna jer omogućava kasniju analizu programa pomoću *codeview* dibagera. Primećujemo da je ova linija bloka uvučena, što je pravilo za sve linije iza prve. To je signal programu NMAKE da se još uvek radi o komandnim linijama započetog bloka.

Nakon pročitanog prvog bloka, program NMAKE zna kako treba da formira *izbor.exe*. Međutim, odmah se postavlja pitanje kako se formiraju neophodne ulazne datoteke *izbor.obj*, *b1.lib* i *b2.lib*. Zato moramo navesti odgovarajuće ulazne blokove odmah iza prvog, kao što se u našem primeru i vidi. Biblioteke *b1.lib* i *b2.lib* formiraju se iz odgovarajućih objektnih datoteka, pozivom programa *lib* (pre toga se stara verzija biblioteke jednostavno briše, ako postoji). Što se tiče datoteke *izbor.obj*, kao i preostalih objektnih datoteka, stvar je prilično jednostavna. Svaka objektna datoteka zavisi samo od svoje izvorne verzije (na primer *izbor.obj* i *izbor.c*), a formira se pozivom kompajlera sa opcijom /c (bez linkovanja). Opcija /AL (izbor memorijskog modela *large*) se preporučuje jer je vrlo izvesno da će program premašiti 64 kilobajta čim smo se opredelili da ga razvijamo u modulima. Opcije /Od /Zi nisu neophodne, ali bez njih nećemo moći da koristimo *codeview* dibager.

Na taj način, 11 blokova naše MAKEFILE datoteke opisuje 11 datoteka koje treba formirati na putu ka izvršnoj verziji programa. Celu brigu oko toga dalje preuzima program NMAKE, a naš posao svodi se na naizmenično pozivanje editora (da bismo izmenili nešto u izvornim tekstovima) i zatim NMAKE (da bismo dobili izvršnu i sve usputne datoteke). Naravno, možemo editovati nekoliko izvornih datoteka, pa tek na kraju pozvati NMAKE:

```
edit choice.c
edit console.c
nmake
```

Podrazumeva se da pri radu sa programom NMAKE moramo voditi računa o tačnom vremenu na našem računaru, jer je to jedini kriterijum koji program koristi da bi proverio da li je neka datoteka u međuvremenu menjana. Zato pre početka rada obavezno moramo proveriti sistemski datum i vreme (*date* i *time* iz DOS-a).

ŠTA DALJE

Listing koji objavljujemo mogao bi se lako prilagoditi bilo kom složenom projektu organizovanom na principu menija. U tom cilju, uopšte nije potrebno menjati sam program, već jedino module koji sadrže podatke o menijima. Recept bi se sastojao u sledećem:

1. Ukloniti četiri poslednja modula iz listinga.
2. Napisati onoliko novih modula koliko ima menija i podmenija u projektu. Svaki meni ima svoj naslov, nazive stavki i odgovarajuće procedure.
3. Sve nove funkcije dopisati u listu deklaracija u okviru *datadef.h* i prilagoditi datoteku MAKEFILE.

Naravno, samo razvijanjem sistema menija još uvek nismo napisali program koji radi nešto konkretno. Pojedine stavke u menijima se ne granaju dalje, već zahtevaju nekakve izvršne programe. Takve programe treba napisati kao izdvojene module i dopisati ih u datoteku MAKEFILE. Šablon za to, nažalost, nemamo.

NASTAVAK SA STRANE 48

logično ponuđenim izborima, tako da i korisnik bez većeg poznavanja materije može napraviti zanimljivu aplikaciju. Zaista, posle svega par minuta smo napravili aplikaciju za rad sa datotekama koje je imala pravi *toolbar*, help sa objašnjenjima za sve opcije menija, pravi editorski status bar i tako dalje.

No, po izrađenoj aplikaciji pomoću **AppExpert** sistema, dočekalo nas je novo iznenađenje – **ClassExpert**. Ovo je sjajan alat za kreiranje novih klasa nasleđivanjem postojećih, navigaciju kroz postojeće klase, proučavanje sorsa klase, *Windows* poruka koje se obrađuju, komandi i tako dalje. Prozor **ClassExpert** a se sastoji iz tri površine: **Classes** koja nudi pogled na spisak klasa koje se u aplikaciji koriste, **Events** koji omogućava istraživanje poruka i virtuelnih funkcija i **Edit** deo koji je zapravo standardan editorski prozor. Svaka od površina ima i svoje **Speed** menije koji donose nove mogućnosti, tako da nam se čini da je ovo pravi pogodak. **ClassExpert** omogućava pregled svih klasa aplikacije uz samo par klikova mišem, dodavanje funkcija za obradu željenih poruka u sam sors pritiskom na dugme, nasleđivanje postojećih klasa i formiranje svih aspekata nove klase opet prostim klikovima mišem, i tako dalje. Izuzetno moćna stvar.

Uglavnom, prostor nam ne omogućava da dalje govorimo o OWL 2.0 biblioteci. Ako je i bilo nekih sumnji oko verzije 1.0, ovaj put smo potpuno sigurni da je ovo pravi način da se programira za *Windows* sistem. Nove ekstenzivna klase obuhvataju skoro sve što bi moglo da zatreba i omogućavaju zaista laku (ili lakšu?) borbu sa *Windows API* funkcijama. Aplikacije rađene sa OWL 2.0 će izgledati bolje, ponašati se bolje, biti sigurnije i lakše prenosive od bilo koje druge aplikacije rađene u sorsu, barem u odnosu na veličinu uloženog korisnikovog truda. Ovo je svakako budućnost.

I SVE OSTALO

Naravno, mnogo toga nismo stigli da pomenemo. *Borland C++ 4.0* osim IDE okoline o kojoj smo uglavnom govorili, obuhvata i mnoge druge programe i preko dvanaest (!) raznih *Windows* help datoteka. Tu su poznati *Resource Workshop* koji je dodatno ali ne previše poboljšan, debager za DOS, debager za *Windows* (ovaj put eksterni, za one koji su se na njega navikli u prethodnoj verziji paketa), debager za 32-bitne aplikacije, *WinSpector* alat za prikupljanje informacija o padu aplikacije i tako dalje. Tu su i stari *TLink* i *Make*, pa i *EasyWin* sistem koji je pokušao u prethodnoj verziji da ubedi korisnike da mogu lako da pređu na *Windwos* programiranje, u čemu definitivno nije uspeo, i u ovoj verziji se povukao u senku. Konačno, tu je i čitav niz malih alata koji već uobičajeno stižu sa svakim *Borland C++* paketom.

Nismo pomenuli ni sjajan **Help** sistem, koji je još u prethodnoj verziji paketa briljirao, a sada je doveden do savršenstva. Tu je, da ne zaboravimo, izvanredan **Master Index**, help datoteka koja će vam pomoći da u dokumentaciji nađete detaljno objašnjenje za bilo šta što vam padne na pamet. Nismo pomenuli ni **Tool** podmeni IDE okoline koji vam omogućava pozivanje bilo koje eksterne aplikacije, kao ni mogućnost podešavanja *Toolbara*. Nismo stigli ni do mogućnosti korišćenja *Visual Basic* proizvoda u kombinaciji sa OWL bibliotekom.

EPILOG

Šta reći na kraju ovog prikaza? Moramo priznati da smo, sa jedne strane, prijatno iznenađeni napretkom u IDE okolini, postojanjem integrisanog debagera, prerađenom OWL bibliotekom, savršenim help sistemom. Sa druge strane, malo smo razočarani slabim pokušajem vizuelnog pristupa programiranju koga će sledeći *Visual C++* sasvim sigurno nadmašiti, kao i izuzetno neprijatno rešenim savlađivanjem razlika između OWL 1.0 i OWL 2.0 biblioteka. IDE okoline dalеko više opterećuje sistem nego prethodna, a na ovim našim prostorima je to i te kako važna činjenica. Mi bismo rekli da je minimalna konfiguracija za iole ozbiljniji rad 486 na 50MHz sa 8MB memorije i lokalnom magistralom.

Odlučićemo se da ovaj paket ocenimo odličnom ocenom i preporučimo ga svakom ozbiljnom programeru za *Windows* okolinu. Paket ima sve što je potrebno, čak i previše, i investicija uložena u njega će se sasvim sigurno isplatiti, zahvaljujući kvalitetu proizvoda koji ona po definiciji ume da stvori. Trenutno je ovo svakako vodeći paket ove vrste na tržištu, mada smatramo da će prvi susedni takmac brzo da ga dostigne, bar do sledeće verzije *Borlanda C++*.

SLOŽNA PORODICA

Kroz jedanaest prethodnih tekstova upoznali smo arhitekturu i osnove programiranja mikroprocesora 80486. Ostalo je još da ukratko opišemo zašticeni i virtuelni režim rada i upoznamo važnije razlike među pojedinim članovima slavne 80x86 porodice.

Dejan Ristanović

Zaštićeni režim (*protected mode*) aktivira se setovanjem bita najmanje težine u kontrolnom registru 0 (CR0). Treba, ipak, znati da se tom setovanju ne sme tek tako pristupiti – ako flег PE „na divlje" postavite na jedinicu, 80486 će raditi u zaštićenom modu veoma kratko, tačnije do prvog interapta, FAR skoka ili promene vrednosti nekog od segmentnih registara. U tom trenutku procesoru će zatrebati tabela deskriptora i pošto ona nije inicijalizovana, dolazi do pada sistema. Pre postavljanja PE treba, dakle, pripremiti osnovne strukture podataka.

ZAŠTIĆENI MOD

Ne moraju se, srećom, inicijalizovati baš sve strukture koje smo u ovim tekstovima pominjali: za prvu ruku dovoljni su globalna tabela deskriptora (GDT) i tabela deskriptora koji opisuju interapte (IDT), naravno uz kreiranje TSS-a (*Task State Segment*) bar jednog procesa. Inicijalna stanja tih tabela mogu se nalaziti u ROM-u, ali ih pre prelaska u zaštićeni režim svakako treba prekopirati u RAM jer ih 80486 u raznim situacijama automatski menja, nemogućnost neke od tih promena automatski znači krah sistema. Pošto su tabele prenesene u RAM, treba podesiti registre GDTR i IDTR da ukazuju na njih i... sve je spremno.

Slika 1 prikazuje najjednostavniju inicijalnu globalnu tabelu deskriptora. GDT[0] se ne koristi, GDT[1] ukazuje na samu tabelu kako bi operativni sistem po potrebi mogao da dodaje i menja deskriptore, GDT[2] ukazuje na interapt tabelu deskriptora (IDT), GDT[3] na TSS prvog (i za sada jedinog) procesa u sistemu, GDT[4] na segment podataka tog procesa, a GDT[5] na kod segment, odnosno program koji će se prvi izvršiti. Na slici 1 taj program je smešten u ROM, što će obično i biti slučaj – program će zapravo završiti inicijalizaciju sistema i kasnije pokrenuti druge programe. Moglo bi se zamisliti i rešenje u kome bi se kompletna inicijalizacija obavila u realnom modu, pa bi se tek onda prešlo u zaštićeni režim i odmah krenulo sa opsluživanjem korisnika. Inicijalni program bi se u tom slučaju verovatno nalazio u RAM-u u koga bi bio učitan sa diska. Preporučljivo je, međutim, što pre preći u zaštićeni režim i iz njega konfigurisati sistem, jer u zaštićenom režimu sam mikroprocesor pruža značajnu pomoć pri pronalaženju eventualnih bagova. Što su alatke ranije pristupačne, to je i razvoj lakši.

Sam prelazak u zaštićeni režim protiče otprilike ovako: u GDTR se upisuje adresa i limit GDT, IDT preko odgovarajuće kapije (koju treba kreirati) ukazuje na rutine koje obrađuju pojedine interapte i greške, IDTR ukazuje na IDT, u TR se stavlje selektor GDT[3], setuje se PE bit registra CR0 i vrši FAR JUMP instrukcija koja će upisati odgovarajući deskriptor u registar CS. Tako startovani program će najpre kreirati svoj segment steka i podataka, definisati stek pointer, popuniti sve tabele operativnog sistema, eventualno pokrenuti mehanizam memorijskih stranica i krenuti sa izvršavanjem korisnikovih zahteva.

Šta da se radi ako se iz zaštićenog režima treba vratiti u realni? Najpre se deaktivira stranične memorije skokom na neku rutinu čija je virtuelna adresa jednaka fizičkoj i obriše PG bit registra CR0. Zatim treba upisati nulu u registar CR3, čime se briše kompletan PDBR (*page directory base register*) i prazni TLB. Veoma je važno da sve ovo obavi program smešten na fizičkoj adresi koja je jednaka virtuelnoj, jer će u protivnom 80486, pošto je stranična memorija deaktivirana, početi da izvršava neki kod na „slučajnom" mestu, što će verovatno rezultirati totalnim krahom sistema.

Sledi postavljanje atributa deskriptora svih segmenata na vrednosti kompatibilne sa realnim modom: granularnost se mora postaviti na „bajt", limit na 0FFFFh (64 kilobajta), a B i D bitovi na nulu. CS mora da dobije atribut *executable*, a SS, DS, ES, FS i GS atribut *writable*. Sada treba onemogućiti interapte, upisati u IDTR baznu adresu 0 i limit 3FFh (vektori 8086 su na početku adresne mape), obrisati PE bit registra CR0 i izvršiti jedan FAR JUMP kako bi se ispraznio red za pripremu instrukcija i eventualno korigovale vrednost CS registra. Ostaje još da se definišu stek pointer i ostali registri i nastavi se izvršavanjem programa u realnom režimu.

Čitava ova procedura je, kao što se vidi, veoma komplikovana, spora i pomalo rizična – preporučujemo da se u realni mod nikada ne vraćate. Ako želite da izvršavate programe pisane za 8086, najbolje je da se oslonite na mehanizme ugrađene u sam 80486, pre svega na virtuelni mod.

VIRTUELNI MOD

Suština virtuelnog moda je kreiranje većeg broja „8086 procesora" koji rade potpuno nezavisno jedan od drugoga i „nesvesni" jedan drugoga, ali pod kontrolom operativnog sistema koji koordinira njihovim izvršavanjem, što uključuje i mogućnost zaustavljanja eventualno „zalutalih" procesa. Treba zapravo reći da je virtuelni mod neka vrsta Intelovog „popravnog ispita": mikroprocesor 80286 napravljen je tako da realni i zaštićeni režim budu u priličnoj meri odvojeni. Intel se nadao da će postojeći DOS programi brzo „migrirati" u zaštićeni režim i da će realni ubrzo postati gotovo nepotreban. To se, iz raznih razloga, nije dogodilo, pa je 80286 neprekidno radio kao malo ubrzani 8086, a zaštićeni režim je malo ko koristio. Intelova poslednja odstupnica, takozvani dualni mod u kome se pišu programi koji mogu da rade kako u realnom tako i u zaštićenom režimu (isti .EXE fajl) totalno je ignorisana od strane proizvođača softvera, pa nije bilo druge nego da se prizna „greška" i napravi novi procesor koji će obezbediti koegzistenciju više programa koji imaju utisak da rade u realnom režimu: taj procesor nazvan je 80386, a veoma sličan model primenjen je kod 80486 i kod Pentiuma.

Operativni sistem kreira proces koji se izvršava u virtuelnom režimu (V86) tako što setuje polje VM kopije registra EFLAGS koja se nalazi u TSS-u tog procesa. Čim se proces startuje a EFLAGS napuni takvim sadržajem, prestaje upotreba deskriptora: bazne adrese se dobijaju množenjem vrednosti selektora sa 16 (tj. šiftovanjem). Reklo bi se da je sve identično sa realnim režimom, ali postoji jedna izuzetno važna razlika. U realnom režimu sve adrese su fizičke adrese, dok u V86 modu operišemo sa linearnim adresama koje odgovarajući hardver može mapirati u skladu sa straničnom organizacijom memorije. To praktično znači da, premda program koji se izvršava ima utisak da mu je na raspolaganju samo megabajt memorije, taj megabajt može biti bilo gde, a njegova mapa se u toku rada može menjati tako da program ima pristup znatno većem adresnom prostoru od 4 gigabajta – na taj način se, na primer, ostvaruje emulacija EMS (LIM) memorije na 80386 i 80486 računarima. Zapravo, čak i ako radite pod „čistim" DOS-om, vaš PC je vrlo verovatno u virtuelnom režimu u koji ga je postavio EMM386, QEMM ili neki drugi program za upravljanje memorijom.

Pristupom memoriji se, dakle, lako upravlja mehanizmima ugrađenim u sam 80486, ali bi pristup periferijskim uređajima mogao da bude problematičan – zamislite kakav bi haos nastao kada bi dva V86 programa „nesvesna" jedan drugog počela istovremeno da šalju znakove na štampač! Takvi problemi se rešavaju pomoću I/O privilegija (na 80486 IOPL, *input-output privilege level*) i IOPB (*input-output permission bit map*).

IOPL je, da se podsetimo, polje koje određuje granicu između prstenova koji mogu da izvrše instrukcije CLI (zabrana prekida), STI (omogućava prekide) i „srodne" LOCK, POPF, PUSHF... – ako je IOPL=3 svi procesi će moći da onemogućavaju prekide, dok će, na primer, IOPL=2 učiniti IF fleg pristupačnim samo procesima u nultom, prvom i drugom prstenu. Pošto se svi V86 programi izvršavaju u trećem (najmanje privilegovanom) prstenu, na ovaj način možete ih sprečiti da onemoguće prekide. I to sa veoma jakim razlogom: program koji bi uspešno izvršio CLI a zatim upao u mrtvu petlju mogao bi se prekinuti jedino resetovanjem kompletnog računara! Pokazuje se, zapravo, da je potpuno nemoguće napraviti pouzdan višeprogramski operativni sistem koji bi podrazumevao IOPL=3 – savetujemo vam da zabranu i ponovno omogućavanje prekida poverite isključivo nekoj rutini iz nultog prstena tj. jezgru operativnog sistema. S ob-



Najjednostavnija inicijalna globalna tabela deskriptora

Razlike između 8086 i 80386/486 u realnom režimu:

- 8086 ne generiše interapte 6, 8, 13, 16 i 17
- na 8086 pokušaj deljenja nulom (*divide fault* INT 0) na steku ostavlja CS:IP koji ukazuju na **sledeću** instrukciju; na 80386/80486 CS:EIP sa steka ukazuje na instrukciju koja je izazvala prekid. Slično tome, vrednost sa steka po „koprocesorskom interaptu" će na 80386/80486 ukazivati na prvi prefiks problematične instrukcije, dok će na 8086 ukazivati na samu instrukciju
- Instrukcija PUSH SP na 8086 je „postdekramentna", a na kasnijim procesorima „predekramentna". To praktično znači da će 8086 najpre baciti na stek adresu pa onda umanjiti SP, dok će 80386/80486 najpre umanjiti pointer pa onda poslati vrednost na stek. Ovo se često koristi u rutinama koje određuju tip procesora; ako negde treba simulirati 8086 ponašanje, umesto PUSH SP koristi se PUSH BP / MOV BP,SP / XCHG BP,[BP]
- ni jedna 80386/80486 instrukcija ne sme da bude duža od 15 bajta. Duže instrukcije, naravno, ne postoje ni na 8086, ali se nizom redundantnih prefiksa tamo može sintetisati instrukcija gotovo neograničene dužine koje će na kraju korektno raditi. Na 80486 ovako duga instrukcija izaziva *general protection fault*
- prefiks LOCK se kod 80386/80486 može primeniti samo na neke instrukcije, o čemu je bilo reči. Kod 8086 ovaj prefiks teorijski može da se primeni na svaku instrukciju, ali kod nekih očito nema smisla
- 8086 će izazvati *divide fault* ako se pokuša deljenje (po apsolutnoj vrednosti) najvećim negativnim brojem (IDIV). Na kasnijim procesorima ovaj bag je otklonjen
- Nastupanje nemaskiranog interapta će na 80386/80486 automatski zabraniti sve prekide, uključujući i nemaskirane. Na 8086 ovaj mehanizam ne funkcioniše, pa u toku obrade nemaskiranog prekida **može** da nastupi novi nemaskirani prekid
- Ni 8086 ni njegovi naslednici ne dopuštaju da se operand prostire preko granice segmenta, ali je ponašanje u konfliktnim situacijama različito. Ako 80386/80486 izvrši instrukciju sa adrese 65535 koja zahteva sledeće bajtove kao operande, nastupiće *general protection fault*. 8086 će naprosto „premotati" adresni prostor i preuzeti operand sa adrese 0. Slično važi i za stek
- Instrukcije FENI i FDISI mikroprocesora 8086 (odnosno koprocesora 8087) su na 80387/80486 NOP
- 8087 ne vrši automatsku normalizaciju racionalnih brojeva – generiše se *denormal exception* i očekuje da aplikacija da izvrši potrebnu normalizaciju. Na 80387/80486 ovu funkciju je preuzeo hardver
- 8087 zahteva WAIT instrukcije pre svake naredbe koja operiše sa racionalnim brojevima, kako bi se uskladilo izvršavanje naredbi procesora i koprocesora. Na 80387/80486 ove instrukcije nisu neophodne, ali i ne smetaju

Razlike između 80286 i 80386/80486:

- Na 80286 LOCK prefiks označava zaključavanje čitave memorije tokom izvršavanja sledeće instrukcije. Na 80386/80486 zaključava se samo memorijska celija kojoj se pristupa
- Stanje mnogih registara 80286 posle reseta je nedefinisano. Na kasnijim procesorima ovi registri su, kao što smo prošloga meseca videli, dobili odgovarajuće početne vrednosti
- 16 nižih bitova registra CR0 na 80286 inicijalno imaju vrednost 0FFF0h, dok su na 80386/80486 u startu 0
- Programi u zaštićenom režimu 80286 izvršavaju se na 80386/80486 ukoliko je četvrta reč svakog od deskriptora postavljena na nulu. 80286 nije koristio ovu reč, pa njeno stanje nije bilo bitno, ako je neki programer i pored preporuke proizvođača propustio da je inicijalizuje, na 80386/80486 može da bude problema
- FSETPM instrukcija 80286 (odnosno 80287) na 80387/80486 je NOP

Razlike između 80386 i 80486:

- 80486 će (opciono) generisati *alignment fault* kod bilo kog pristupa memoriji koji operiše sa dva ili više bajtova
- Korišćenje nekih od 80486 flegova ili kontrolnih registara na 80386 izazvaće odgovarajući *fault*
- Jedinica za rad sa racionalnim brojevima mikroprocesora 80486 (takozvani NDP ili popularno „80487") ne generiše interapt 9; umesto njega generiše se interapt 13. O ovome je bilo reči kada smo opisivali interapt mehanizam 80486
- NDP mikroprocesora 80486 obezbeđuje „redirekciju" izveštaja o greškama preko NE bita registra CR0 i pinova FERR\ i IGNNE\

zirom da programi u trećem prstenu ne mogu da promene vrednost IOPL, takvo ograničenje je po prirodi stvari permanentno.

Kada se radi o pristupu periferiji, ne može se biti tako restriktivan – najzad, i siroti V86 programi moraju nešto da štampaju, a da i ne govorimo o pristupu disku, tastaturi ili ekranu. Srećom, konstruktori 80486 predvideli su u TSS-u svakog procesa polja zvana *I/O permission bits* – svaki bit odgovara jednom od portova, pa se svakom pojedinačnom programu može dopustiti ili zabraniti pristup svakom konkretnom periferijskom uređaju. Ukoliko je pristup tom uređaju zabranjen, svaka IN i OUT instrukcija izazvaće interapt 13, tzv. *general protection fault*.

KONTROLA PRISTUPA

Autor operativnog sistema može da se opredeli za različite strategije kontrole pristupa periferiji. Može pre svega uređaje dodeljivati statički: pri kreiranju procesa od korisnika se zahteva da definiše prava pristupa i onda se te odluke ugrađuju u mapu TSS-a; ako korisnik dopusti da dva procesa pristupaju štampaču pa nastane haos, korisnik je sam kriv! Jasno je da se ovakvo rešenje može primeniti tek na najjednostavnijim operativnim sistemima.

Najpotpunije rešenje nalaže kompletnu zabranu pristupa periferiji za sve procese. Svaki pokušaj izazvaće interapt 13, da bi se onda rutina koja obrađuje taj prekid „snalazila": proveriće da li je uređaj slobodan; ako jeste, dodeliti ga procesu, ako nije, odložiti njegov zahtev u neki red za čekanje i tome slično. Ovakvo rešenje je nešto složenije za realizaciju, ali to nije jedini problem sa njim – potreba da se kod svake IN ili OUT instrukcije pozove jedan interapt i izvrši relativno komplikovana procedura znatno usporava komunikaciju sa uređajima. To možda i nije toliko važno kada se radi o nekom sporom štampaču ili skeneru, ali bi moglo da predstavlja pravu katastrofu ako se pristupa ekranu ili nekom brzom medijumu masovne memorije. Zato se uglavnom pribegava kompromisnom rešenju: u startu je svim procesima zabranjen pristup kompletnoj periferiji. Onoga momenta kada neki proces **prvi put** izvrši neku IN ili OUT instrukciju, izvršiće se INT 13 koji će proveriti da li je željeni uređaj slobodan. Ako jeste, postaviće odgovarajući bit u TSS-u datog procesa kome se daje ovlašćenje za direktan pristup uređaju, pa će buduće IN i OUT instrukcije raditi sasvim normalno. Naravno, pokušaj nekog drugog procesa da pristupi istom uređaju biće onemogućen, a proces suspendovan ili čak prekinut. Glavna mana ovog rešenja je što nije lako odlučiti kada se neki uređaj može „oduzeti" procesu i proglasiti slobodnim. DOS, na žalost, ne podrazumeva nikakve mehanizme alokacije ili oslobađanja uređaja i uvek misli da je sve na sistemu baš njegovo.

Obrada prekida u virtuelnom režimu predstavlja veoma osetljivo pitanje, pre svega zbog dualne prirode ovoga moda. Sa jedne strane, izvor interapta može da bude sam program koji će, na primer, izvršiti neku INT instrukciju i očekivati da ona bude obrađena od strane njegovog *interrupt handler*-a (sećamo se da programi u V86 modu „pojma nemaju" da nisu sami na sistemu). Sa druge strane, izvor interapta može da bude nešto van virtuelnog procesa, na primer pomenuti *general protection fault*, u kom slučaju obradu treba da preuzme jezgro operativnog sistema ili neka rutina iz nultog prstena. Prekinuti program u tom slučaju neće ni znati šta ga je zadesilo.

Mehanizam obrade prekida u V86 modu principijelno se ne razlikuje od obrade prekida u zaštićenom režimu: preko odgovarajućeg deskriptora IDT a kontrola će biti predata odgovarajućoj rutini iz nultog prstena, adresa povratka stavlja se na stek nultog prstena (na njega ukazuje TSS), a ne na stek tekuće aplikacije. Hardverski interapti se na nulti prsten usmeravaju preko odgovarajuće kapije, dok je kod softverskih prekida situacija nešto složenija – operativni sistem mora da odredi izvor prekida da bi ga na odgovarajući način obradio.

Većina softverskih prekida pokušaće skok na rutinu operativnog sistema preko 8086 kompatibilnog vektora, što će, kao i svaki drugi poziv rutine iz različitog prstena, trenutno izazvati interapt 13, *general protection fault*. Program koji obrađuje ovaj prekid mora da ispita kod greške sa steka i tako ustanovi koji je interapt zapravo bio pozvan – ako je to neki od klasičnih 8086 prekida i ako se ustanovi da je izvor neka INT instrukcija, biće podešen stek korisnikove aplikacije i kontrola vraćena njenom *interrupt handler*-u. Ukoliko je izvor prekida nešto drugo, treba preduzeti odgovarajuće akcije i vratiti kontrolu korisnikovom programu bez izvršavanja njegove interapt rutine.

KOMPATIBILNOST...UZ RAZLIKE

Intelovu 80x86 porodicu krasi visok stepen vertikalne kompatibilnosti: svaki „sledeći" procesor sadržao je „u sebi" sve prethodne, pa su se programi mogli bez ikakvog ograničenja prenositi sa računara na računar. To ipak ne znači da između pojedinih Intelovih mikroprocesora nema i određenih razlika koja istina ne prave veće smetnje, ali koje u nekim situacijama ipak mogu da budu dekako bitne. O najvažnijim razlikama, na primer A20 liniji koja je odgovorna za *High Memory Area* DOS-a 5.0, već je bilo reči; ovde ćemo pomenuti samo neke sitnije (?) izmene.

Ovim smo došli do kraja naše serije tekstova posvećene mikroprocesoru 80486 – u pravo vreme, reklo bi se, pošto je Pentium iz dana u dan sve popularniji. Srećom, velika većina ovde izloženih informacija važi i za Pentium, a verovatno će važiti i za neke od budućih Intelovih mikroprocesora iz ove porodice. U sledećim „Računarima" ćemo, na zahtev velikog broja čitalaca, objaviti svojevrsnu rekapitulaciju ove serije tekstova: pokušaćemo da, bez ulaženja u detalje, izložimo osnovu arhitekture 80486 i njenog asemblera. Takođe ćemo dati spisak korišćene literature – konsultacija knjiga i priručnika koje ćemo pomenuti pravi je način da produbite i primenite znanja koja ste stekli čitajući „Računare"!

ČUVANJE PODATAKA

I pored velikog broja paketa koji služe za čuvanje podataka, uvek mi je nedostajala poneka biblioteka koja bi bila i zanimljiva i laka i slobodna za upotrebu, pa mi je jedino preostajalo da sam napišem ono što mi je potrebno. Ovde predstavljam skup funkcija za rad sa memo-poljima, odnosno mali model baze koja omogućava čuvanje podataka u nekom uređenom formatu.

Miljan Jovanović

Ovde će biti reči o još jednoj grupi funkcija za čuvanje podataka. Možda će neko reći da izmišljam toplu vodu, ali do sada jednostavno nisam naišao na dovoljno zanimljivih biblioteka za laku i slobodnu upotrebu. Svi paketi koje sam imao prilike da vidim imali su poneku manu zbog koje sam rešavao da sve što mi je potrebno napišem sam. Najčešće su u pitanju bile loše koncepcije, previše nepotrebnog koda, loše uputstvo itd. Ipak, od svega me najviše odbija nedostatak izvornog koda. Doduše, neke stvari zaista delu ju mnogo bolje kao su upakovane u lib ili exe ali ipak je lepše imati originel koji se po potrebi može bar malo „doterati". Uvek kada vidim da je izvorni kod isporučen kao deo paketa pomislim „Jaooo šta ću sad da napravim!" i na tome se sve završi. Jer „provaliti" nečiji stil pisanja i još važnije koncepciju programa zahteva toliko vremena da se na kraju sve svodi na „uzmi ili ostavi". U mom slučaju ono drugo je mnogo češće.

Dugo sam imao dilemu da li treba u javno vlasništvo stavljati i izvorni kod. Nakon nekoliko godine programiranja i druženja sa programerima zaključio sam da programe ne treba skrivati plašeći se suda drugih. Ako vrede, nagrada će stići sigurno, a ako nešto nije baš najbolje, brzo ćete dobiti kritike i sugestije.

Na kraju ovog uvoda želim da kažem i to da sam jezik C naučio čitajući knjige (i to pre no što sam nabavio računar), želim iz razgovora sa prijateljima i tumačenjem raznih izvornih kodova. Ovo poslednje su mi omogućili mnogi naši programeri (uglavnom saradnici „Računara" ili članovi Sezama), kao i oni širom sveta, pa im se ovom prilikom zahvaljujem.

MODEL BAZE

Prilikom izrade svog prvog većeg projekta, naišao sam na probleme koji su isprva delovali trivijalno, ali se pokazalo da naoko „jednostavan" posao zahteva dosta vremena. Uglavnom se to odnosilo na sistem menija i uopšte vezu sa korisnikom. Ono što me je činilo srećnim je da sam „iz glave" u jednom dahu napravio sistem za čuvanje podataka sa indeksima i ostalim stvarima. Brzina nije bila baš velika, ali ipak je to bio uspeh.

Ovde ću opisati mali skup funkcija za rad sa *memo* poljima, ali moram naglasiti da se ne radi ni o čemu revolucionarnom već samo o jednostavnom primenljivom rešenju. Radi se o malom modelu baze koja treba da omogući čuvanje podataka u nekom uređenom formatu. Što se tiče indeksa, to ostavljam vama da sami napravite po sopstvenom ukusu („Računari 83", str. 41, Dejan Jelović „Indeksiranje baze – B stabla").

Memo polja je polja u okviru sloga koje nije ograničeno dužinom (uslovno) i koje se ne koristi za smeštanje podataka potrebnih za sortiranje/indeksiranje baze. Zbog „neograničene" dužine ovo polje se čuva odvojeno od sloga, u posebnoj datoteci, a kao veza, koristi se neki *id* broj. Ono čime ćemo se mi ovde baviti je kako organizovati datoteku sa *memo* poljima. Ovaj problem se može vrlo jednostavno rešiti: otvorićemo datoteku, u nju upisati potrebno, zapamtiti mesto i dužinu podatke i – to je to! Kasnije, ako nam je potrebno neko polje, lako ćemo ga naći u šumi sličnih. Te datoteke će iz dana u dan biti sve veća, a kad dođe vreme da se iz nje izbace nepotrebne stvari javiće se problem! Za fizičko uklanjanje podataka biće potrebno otvoriti još jednu takvu datoteku u koju ćemo prepisivati samo ona polje koja treba zadržati. Ako smatramo da će biti dovoljno mesta na disku onda je sve u najboljem redu, ali to najčešće nije slučaj.

ČLANKOVITA DATOTEKA

Drugacija koncepcija postoji odavno i koristi se skoro redovno za čuvanje podataka. Potrebno je samo datoteku „izdeliti" na jednake delova (sektore) tako da brisanjem sadržaja sektora on postane slobodan za nove informacije. Ovo rešenje sam imao prilike da vidim na delu koje je napisao Zoran Životić (Sezam Off-line Reader) i priznajem da mi se veoma dopalo. Dakle, ideja je dosta stara i vrlo brzo sam je projektovao na svoje probleme – došao sam do zaključka da i meni baš to treba. Hteo sam da napravim sistem za rad sa bazom koji će biti isti kao i sistem za rad sa diskom (direktorijumi i datoteke), ali da se sve čuva u jednoj datoteci.

Prvo sam napravio neke testove i došao do zaključka da u zahtevima treba biti skromniji – mnogo skromniji. Izmerene vrednosti posle analize dovele su me do zaključka da je najoptimalnije sledeće rešenje: jedne sektorisana (člankovita) datoteka koja počinje informacijama od opšteg značaja za celu bazu, zatim sledi tabela o slobodnim sektorima i na kraju sami sektori. Stvarno se ovo pokazalo kao dovoljno brzo i upotrebljivo. Čak je i kod ostao pregledan i razumljiv – gotovo nikakvo opterećenje za programera. Tokom projektovanja cilj mi je bio da se svi podaci čuvaju u jednoj datoteci koja bi imala tri celine:

1. **Zaglavlje baze** – dužine 128 bajtova. Prva dva bajta su „potpis" da se zna da se radi o bazi (dva bajta). Ovaj potpis se testira prilikom otvaranja, ako to ne prođe, datoteka neće biti otvorena. Ovo sam dodao tek da bih onemogućio slučajan pristup pogrešnoj datoteci – bilo kakav pristup (pisanje) pogrešanoj datoteci je poguban za nju.
2. **Tabela alokacija** – „okrugle" dužine od 8k. Svaki bit predstavlja po jedan sektor i nosi informaciju o tome da li je on zauzet ili ne.
3. **Sektori** – podaci su smešteni u sektore koji počinju zaglavljem sektora. Dužina sektora se određuje prilikom kreiranja datoteke, a taj podatak se nalazi u zaglavlju baze (datoteke sa MEMO poljima).

Veliki problem predstavlja ograničenja DOS-a: u jednom prolazu može se pisati/čitati samo blok do 64kb. Ako je pak podatak duži, onda treba uzastopno pozivati funkciju **fwrite** (i naravno, u suprotnom **fread**) tako da se pokrije ceo niz. Odlučio sam se da maksimalna dužina memo polja bude ispod te barijere iz nekoliko razloga, od kojih izdvajam najvažniji: nisam imao potrebe za obradom većih podataka. Mislim da nije veliki problem prepraviti funkcije za čitanje/pisanje tako da se i ovo omogući.

Zaglavlje baze

U zaglavlje baze treba smeštati samo podatke vezane za celu bazu. Za ovaj deo rezervisano je 128 bajtova, od kojih je zauzeto svega 5 – ostalo je rezervisano za buduća proširenja. Prva dva bajta sam već opisao – potpis baze, zatim sledi podatak o veličini sektora. Taj podatak je zapisan u jednom bajtu i zapravo predstavlja stepen dvojke. Očekivana dužina sektora je u rasponu od 128 bajtova do 8k, mada je moguće zadati i maksimalnu vrednost od 2^{256}. Ovo će izazvati goleme probleme, pa o tome treba voditi računa (pak zbog DOS-ovog ograničenja najveća dužine sektora je 64k).

Tabela alokacija

Ova tabela nosi slično ime kao i FAT (*File Allocation Table*) i ima sličnu funkciju, ali ne i istu strukturu. Zapravo, radi se o tome da u ovoj tabeli postoje informacije o zauzeću sektora, ali ne i o lancu koji treba da predstavlja jednu logičku celinu. Zašto? Pa, odgovor je vrlo jednostavan: ovo rešenje ne omogućava ulančane sektore! Ulančani sektori su moja druga zabluda vezana za ovaj sistem. Takvo rešenje se u praksi pokazalo kao sporo i vrlo brzo sam ga zaboravio. Drugo, kod je srazmerno problemu bio veći i teži za praćenje – bilo je potrebno voditi računa o mnogim stvarima koje nisu davale očekivane rezultate. Sada se vezani sektori nalaze u nizu jedan za drugim, pa tabela alokacija sadrži samo informaciju da li je sektor slobodan ili ne. Dužina ove tabele je 8k – potrebno za 8·8k sektora (64k), tako da se njihova adresa može smestiti u jedan „unsigned int", što olakšava mnoge probleme vezane za pretragu po tabeli i drugim stvarima.

Sektori

Sektori jednog podatka se nalaze u nizu jedan za drugim – dovoljan je samo jedan poziv funkcije **fread()**. Pre svakog MEMO polja nalazi se zaglavlje sektora sa dužinom podatka koji sledi. Ovo zaglavlje gotovo da nije ni potrebno, ali time se smanjuje verovatnoća pristupa podacima „od sredine". Može se smatrati da je ono tu iz praktičnih razloga – dešavalo mi se da iz pokvarene indeksne datoteke dobijem neispravne brojeve sektora – eto brljotine! Pored „potpisa" kojim zaglavlje mora početi, tu se nalazi i dužina (u bajtovima) podatka koja sledi iza njega.

Kako se veličina sektora zadaje kao stepen dvojke, onda je moguće ovaj podatak čuvati u samo jednom bajtu. To praktično znači da je sektor ograničen veličinom od 2^{255} – vi samo zamislite kolika je to vrednost (prično neupotrebljivo, zar ne?). Realna veličina sektora je od 64 do 64k-1, pa se to sve čuva kao 2^6 do 2^{16}. Pretpostavljam da nije potrebno raditi sa sektorima manjim od 64 bajta, a da su optimalne veličine 256 i 512, no tokom rada sami možete pronaći optimum.

Funkcije i strukture

Kao prvo, za lakši rad uveo sam neke nove tipove promenljivih:

```
typedef unsigned char    byte;
typedef unsigned int     word;
typedef unsigned long    dword;
```

Sasvim je razumljivo šta one predstavljaju, a ja ih koristim u svakom programu koji pišem. I osnovne strukture koje se koriste za grupisanje podataka treba „zaviti" u nove tipove promenljivih (slika 1, 2, 3). Kad god nam zatreba zaglavlje baze, koristićemo BASE_HEAD.

Iz datog primera može se zaključiti da ova struktura po načinu primene ima sličnosti sa **FILE** (izvorni kod se nalazi na Sezamu u direktorijumu R:\RSOFT\). Spisak svih funkcija je dat na slici 4.

Kreiranje, otvaranje i zatvaranje baze

Kreiranje baze se nimalo ne razlikuje od kreiranja bilo koje datoteke – uostalom, sve funkcije se čak zovu slično onima za rad sa datotekama iz standardne biblioteke. Mala je razlika u parametrima koji se prosleđuju, ali su oni logično (?) zadati. Dakle, kreiranje se vrši funkcijom **bcreate()** i njoj se prosleđuju dva parametra: pointer na ime datoteke i dužina sektora. Dužina sektora je ograničena na 64k-1 (65535 bajtova), tako da se taj parametar prosleđuje kao word. Ova funkcija nam vraća pointer na kontrolnu strukturu preko koje ćemo pristupati bazi. Primer bi izgledao ovako:

```
BASE *b;
b = bcreate("memo.b", 512);
```

Time će biti kreirana prazna baza po imenu MEMO.B čiji će sektori biti dužine 512 bajtova. Inicijalno tu će biti zapisano ono što je potrebno za ispravan rad: zaglavlje baze i tabela alokacija prazne baze (popunjena nulama). Pošto se ovom komandom vrši odvajanje memorije za tabelu alokacije i zaglavlja, tre-

Slika 1

```
typedef struct _bhead {
    word Sign;                          potpis baze
    byte SecLen                         dužina sektora, stepen dvojke
    word SSign,                         potpis sektora
    char res[128  2*sizeof(word)  sizeof byte)]     // rezervisano
} BASE_HEAD,
```

Zaglavlje baze

Slika 2

```
typedef struct _shead |
    word SSign                          potpis sektora
    word len                        |   dužina podatak u bajtovima
    char res[16  2*sizeof(word)]|      / rezervisano
} SEC_HEAD
```

Zaglavlje početnog sektora

Slika 3

```
typedef struct _base {
    FILE *file                          id datoteke | file handler)
    BASE_HEAD *BaseH                  | pointer na zaglavlje baze
    SEC_HEAD *SecH                      radni prostor za zaglavlje sektora
    byte *AT                            tabela alokacija
    word SecSize                        dužina sektora (u bajtovima)
    long FstSecOffset,                  pozicija prvog sektora u bazi
| BASE
```

Kontrolna struktura za pristup bazi

Slika 4

```
BASE *bcreate(const char *filename, word SecLe      Kreiranje baze
BASE *bopen(const char *filename)                   Otvaranje baze
int bclose(BASE *stream)                            Zatvaranje baze

int brewind(BASE *stream),                          Skok na prvi sektor
int bseek(BASE *stream  word SecNo  int whence)     Skok na zadati sektor
word btell(BASE *stream)                            Broj tekućeg sektora

word bwrite(void *ptr  word size  BASE *stream      Snimanje podataka u
                                                    bazu
word dlength(BASE *stream, word shead],             Dužina podatka
                                                    upisanog u neki
                                                    sektor
word bread(void *ptr  word SecNo, word Num, BASE *stream]   Čitanje podataka iz
                                                    baze
word bdelete(BASE *stream, word SecNo]              Brisanje podatka iz
                                                    baze
word dmove(BASE *stream  word SecNo],               Premeštanje podatka
                                                    na lokaciju bliže
                                                    početku datoteke

int btruncate(BASE *stream]                         Brisanje nepotrebnih
                                                    sektora
int bclear(BASE *stream]                            Brisanje svih
                                                    podataka iz baze
```

Spisak funkcija za rad sa memo poljima

ba proveriti da li je uspešno obavljen posao. Ako funkcija vrati NULL pointer, znači da je došlo do neke greške tokom alociranja memorije ili otvaranja datoteke. Prilikom kreiranja nove baze, ukoliko eventualno postoji istoimena datoteka, ona će biti obrisana. Ako vam je potrebno samo otvaranje postojeće baze onda primenite funkciju **bopen()**. Ona će, ako baza ne postoji, kreirati novu. Iako de **bcreate()** treba koristiti samo u slučaju isključivog kreiranja nove baze ili za davanja posebne dužine sektora, drugačije od unapred projektovane vrednosti od 512.

Zatvaranje otvorene baze se vrši funkcijom **bclose()**

bclose(b),

Time se pored zatvaranje datoteke vrši i oslobađanje zauzete memorije potrebne za rad ovog sistema. Ako je bilo nekih grešaka, biće vraćena vrednost različita od nule.

Navigacija

Za pozicioniranje na neki slog treba koristiti funkciju **bseek()**. Njeni parametri su slični parametrima funkcije iz standardne biblioteke **fseek()**, jedina razlika je u tome što se offset zadaje u sektorima, a na u bajtovima. Jedan oblik ove funkcije je **brewind()**

int brewind(BASE *stream) {
return(bseek(stream 0, SEEK_SET)),

koja služi za „premotavanje" na početak. Ne verujem da će ova, kao ni funkcija za određivanje tekućeg sektora (**btell()**) naći preveliku primenu, jer ih nivo automatizacije isključuje.

Ulaz / Izlaz

Snimanje podataka u bazu vrši **bwrite()**. Na osnovu zauzeća sektora ona će izabrati *prvo* slobodno mesto najbliže početku (**first fit**) i u **njega upisati sadržaj zadate memorije. Njoj treba proslediti mesto i veličinu podatka koji treba snimiti, a ona će vratiti broj sektora gde su podaci smešteni. U slučaju neke greške, vratiće vrednost veću od *MAX SECTORS*. Izbor sektora vrši funkcija GetFirstFree** (ovu funkciju napisao je Dragan Zakić pre neku godinu...). Na osnovu dužina podatka (zapravo broja potrebnih sektora za čuvanje zaglavlja i podataka), ova funkcija će tražiti u tabeli alokacija „rupu".

Čitanje sadržaja nekog polja se mora odvijati u nekoliko koraka. Prvo saznajemo njegovu dužinu funkcijom **dlength()**, zatim alociramo potrebnu memoriju i u nju pozivom funkcije **bread()** prenosimo sadržaj iz datoteke. **bread()** očekuje sledeće parametre: broj sektora (id broj) početka memo polja i adresom bafera gde taj sadržaj treba upisati (adresa i veličina). Ako dati sektor ne počinje zaglavljem ili ne predstavlja početak podatka ili je bafer mali da prihvati sadržaj celog memo polja, onda će funkcija vratiti nulu.

Povodom alociranja ovog prostora za smeštanje celog polja mogu još da predložim da izmenite funkciju **bread()** (ili da napišete drugu) tako da ona automatski alocira potreban prostor. Ovako bi trebalo da izgleda ta funkcija:

```
char *get memo(BASE *stream  word SecNo)
    byte far *tmp
    word len,
        s..

    len  dlength(stream  SecNo),
    tmp  (byte *)farmalloc(len)
    if(!tmp)
        return(NULL),        // nema dovoljno memorije

    s  bread(tmp, s!  (word)len  stream)
    if(!s)
        return(NULL),        // čitanje nije uspelo
    return(tmp)
```

Brisanje podataka

Kad podatak izgubi vrednost i treba ga ukloniti iz baze, koristićemo funkciju **ddelete()** kojoj prosleđujemo broj sektora gde on počinje. Kako se u zaglavlju svakog sektora nalazi potpis, onda će samo „overen" sektor biti obrisan, a u suprotnom biće vraćeno da je došlo do neke greške. Uklanjanje samo jednog memo polja zasniva se na resetovanju određenih bitova u tabeli alokacija, a ne na fizičko brisanje iz datoteke. Brisanje *svih* podataka se najlakše može ostvariti po-

stavljenjem svih bitova u tabeli alokacija na nulu. Posle **bclear()** baza ostaje otvorena i spremna za nove podatke. Brisanjem poslednjeg podatka iz baze možemo skratiti datoteku na dovoljnu dužinu. Ovaj korak nije neophodan, ali kako ne zahteva previše vremena onda ga posle svakog brisanja možemo ponoviti. Taj postupak je smešten u funkciju **btruncate** a poziva se na kraju **ddelete()**.

Pakovanje baze

Glavna prednost ovog sistema je pakovanje baze. Rešenje ovog problema nije složeno, ali mnogo zavisi od načina indeksiranja pa nema smisla davati neko opšte rešenje. Drugo, potrebe za indeksiranjem su vrlo različite tako da i to izaziva dodatna komplikacije. Ovde ću odvojiti nekoliko pasusa tek da vam dam ideju za realizaciju.

Generalno, treba „pokupiti" sve *id* brojeve memo polja, smestiti ih u neki niz tipa *unsigned int* i sortirati po veličini. U najgorem slučaju za ovo će biti potrebno nešto manje od 128k RAMa. (Naravno, ako je baza puna, onda nije ni potrebno razmišljati o pakovanju.) Zatim krenemo redom po tom nizu i radimo sledeće:

```
for(  )
    uzmi dužinu podatka na osnovu Id broja
    nađi prvo slobodno mesto gde se isti može prebaciti
            (GetFirstFree)
    ako se slobodno mesto nalazi pre tekućeg, onda prebaci podatak i
            upamti njegovu novu adresu, u suprotnom nišla na diraj
    pozicioniraj se na sledeći Id broj iz niza
    ako nema više onda break
```

Promena mesta nekom podatku treba da obezbedi pouzdanost i brzinu. Kako u slučaju nasilnog prestanka rada ne sme doći do gubitka podataka, onda predlažem jedno od „sigurnih rešenja:

1. pročitaj podatak
2. upiši podatak na novu adresu
3. 3ostavi novu adresu na mesto stare (verovatno u nekom indeksu, polju i sl.)
4. obriši stari podatak

Ovim redosledom operacija obezbeđena je sigurnost u svakom trenutku. Zapravo, samo će u slučaju prekida između tačaka 2 i 3 postojati dva kopije od kojih će ona nova biti višak. Princip je vrlo jednostavan i dat je u funkciji **dmove()**.

Sigurno je da postoji više algoritama za pakovanja memo polja, ali ne treba pretarivati u zahtevima. Po pitanju brzine, rad sa diskom je često najkritičnije mesto pa treba izbegavati često čitanje/pisanje podataka. Nije na odmet izabrati neki metod koji će biti brži čak i ako potpuno ne „očisti" datoteku od praznih sektora.

U fazi testiranja u primer sam dodao jednostavan princip indeksiranja. Naravno, radi se samo o pamćenju sektora koji ma počinju podaci. Pakovanje baze se zasniva samo na toj tabeli početnih sektora, a za ozbiljniji rad preporučujem B stabla.

NEPOBEDIVI DRUG IZ KLUPE

Natprosečno inteligentnima obično smatramo osobe sklone aktivnostima u kojima se većina drugih ljudi teško snalazi. Da li zato što su kadri izvršavati složene matematičke operacije i igrati šah na nivou velemajstora, možemo i računare proglasiti inteligentnim? Ipak ne. Jer čak i one poslove u kojima uspevaju parirati čovekovoj kreativnosti, mašine obavljaju na svoj uobičajen način – hladno proračunato!

Miroslav Oleaja



Ideja da se digitalni računar programira za igranje logičkih igara datira još iz najranijih dana razvoja informatike. Pioniri računarske tehnike Alen Tjuring i Klod Šenon prvi su u svojim člancima iz 1950. godine teoretski razradili algoritme na kojima bi se zasnivao jedan program za igranje šaha, već tada je ukazano na probleme sa kojima se i danas suočavamo, a koji se u biti daju svesti na nemogućnost iznalaženja dovoljno efikasnog i pouzdanog kriterijuma za smanjivanje ogromnog broja mogućih nastavaka koje je neophodno analizirati pre povlačenja svakog poteza. Prvi funkcionalni šahovski program načinio je 1957. Aleks Bernštajn na sistemu IBM 704. Mada po snazi skroman i beznadežno spor, vladao je svim pravilima šahovske igre i sa te strane davao korektne odgovore. U godinama koje slede pojavljuje se čitava gama novih elektronskih šahista, sve bržih i sve vispenijih, tako da se u javnosti već počeo stvarati utisak da je blizu momenat kada će računari nadmašiti čoveka, kako u igranju šaha, tako i u svemu ostalom. Tome su posebno doprineli radovi čuvene grupe RAND iz laboratorije u Los Alamosu, u kojima se zagovaralo napuštanje slepog prebrojavanja svih mogućih kombinacija i prelazak na heurističke mehanizme rezonovanja pomoću kojih bi se iz jednostavnih početnih pravila indukovali složeni zaključci. Alan Njuel, Herbert Sajmon i J.C. Šo, koji su sačinjavali pomenutu grupu, uspeli su da svoje algoritme približe ljudskom načinu razmišljanja, ali ne i u njegovom najvažnijem aspektu – podražavanju sposobnosti učenja na greškama i memorisanju i upotrebi ranijih iskustava.

RAZLIKE U MIŠLJENJU

Na koji način čovek igra šah? Šta se to tačno dešava u našoj svesti dok sedimo nagnuti nad crno-bela polja? Ko je ikad sebi postavljao ovakva pitanja, vrlo brzo je shvatio da je razumevanje samih tokova logičkog rezonovanja kudikamo lakše nego razumevanje njegovih ishodišta. Biohemijski procesi u ljudskom mozgu ne samo da su daleko složeniji od računarskih algoritama kakve danas poznajemo, nego se ni na mogu sa njima upoređivati – toliko se po samoj svojoj suštini razlikuju. Računarski algoritam predstavlja niz precizno definisanih koraka koji se sukcesivno izvršavaju po strogo utvrđenom i nepromenljivom redosledu, proces odlučivanja u čoveka je, pak, rezultat poluhaotičnog okidanja nervnih impulsa među par milijardi gusto isprepletanih neurona, od kojih je svaki aktivan (ili čak ni to) istovremeno i gotovo nezavisno u odnosu na druge. Uprkos tome, ili možda baš zbog toga, nervni sistemi čoveka kao i većine životinja sa lakoćom daju ispravna reakcije u okolnostima u kojima se računar nikad ne bi snašao, i izvršavaju zadatke koji su za njega u bukvalnom smislu – misaone imenica.

Uvek kada vam prepuste da nastavite nečiju partiju, potrebno vam je ne malo vremena da 'uđete u situaciju', kako se to obično kaže, odnosno da sagledate kako ko stoji i kakve akcije je moguće preduzeti. Međutim, već nakon par poteza vaše vreme za razmišljanje će se rapidno smanjiti i postepeno približiti onom koje biste i inače trošili da ste igrali partiju od početka. A sad, zamislite da igrate šah tako da posle svakog poteza prelazite na sledeću, uvek novu i nepoznatu poziciju – otprilike kao kad bi igrali simultanku za tolikim brojem tabli da se uopšte ne sećate pozicija iz prethodnog kruga, i da vam se čini kako stalno pred sobom imate nekakvu drugu partiju. Prilično bi mukotrpno bilo tako igrati, a kamoli pobediti protivnika koji sve vreme gleda u isti raspored figura! A upravo bi se ovako mogao opisati položaj u kome se nalazi računar – za njega je nakon svakog povlačenja novodobijena pozicija toliko teško uporediva sa prethodnom, da on praktično nije u stanju da iskoristi zaključke proistekle do tada.

Ne samo da računar nije u stanju da pamti i upotrebi iskustva iz ranije odigranih partija, on nije posebno sposoban ni da iskoristi zaključke do kojih je došao razmatrajući poziciju u prethodnom potezu, recimo tako što bi granu stabla koja počinje potezom koji je protivnik upravo povukao samo dalje produžio, pošto je ta grana tek sićušni deo stabla koje treba nanovo razviti. Iz istog razloga je mala korist od analiziranja pozicija dok se čeka na protivnikov potez, jer računar ima daleko manje šanse nego živi igrač da predvidi koji bi to potez mogao biti.

MILIONI KOMBINACIJA

Bilo da sa računarom igramo šah, mice, go, reversi ili nešto drugo, način njegovog „razmišljanja" je uvek isti. Kad na njega dođe red, on prvo u skladu sa pravilima igre formira listu svih regularnih svojih poteza (poluhodova), potom za svaki od njih napravi novu listu svih mogućih poluhodova protivnika, pa onda opet za svaki od tih novu listu svojih, do dubina koju mu je zadao programer ili, u boljoj varijanti, u zavisnosti od raspoloživog vremena. Međutim, priraštaj ovakvog algoritma je toliki da se računar „zaguši" pokuša li da gleda više od tri-četiri poluhoda unapred. U šahu, na primer, u jednoj prosečnoj poziciji pri središnjici moguće je povući do pedeset različitih poteza. Ako je i protivnik posle svakog od njih u stanju da bira između svojih pedesetak, i još jedanput tako, ta-da bi broj konačnih pozicija koje valja analizirati iznosio tačno ovoliko:

$$50^4 = 6\,250\,000$$

Šest miliona i kusur puta, znači, treba pozivati rutinu za procenu situacije na tabli, da bi se odigrao jedan jedini potez! I to kakav. Odviše kratkovid („ja pa ti, ja pa ti") da bi s njim mogli uočiti „običan" mat u tri poteza. U stvari, prethodna računica je samo hipotetična – nikakav stoni računar i ne može, u roku u od nekoliko sekundi koliko smo spremni čekati ga, generisati ovako veliki broj nastavaka. Niti se to doima nužnim: ni vi kada igrate šah ne razbijate glavu nad više od desetak mogućih kombinacija, pa opet lako izađete na kraj sa svojim kompjuterom.

Da bi u toku procesa donošenja odluka koji potez da odigra umanjio broj kombinacija koje mora ispitati, algoritam ugrađen u program za igranje šaha (kao uostalom i bilo koje druge misaone igre) prinuđen je da prethodno izvrši odabir određenog manjeg broja najperspektivnijih poteza iz skupa svih mogućih u datoj poziciji. Ali, koji postupak primeniti a da se sigurno ne ispusti iz vida onaj uistinu najkvalitetniji? I da li je uopšte moguće definisati neki 100% nepogrešiv kriterijum za preliminarnu selekciju u slučaju tako kompleksne igre kao što je šah? Jer, ako nije, znači li to da uprkos svim megabajtima i megahercima kojima raspolažemo nikad ne možemo biti sigurni da će se doći do zaista najboljeg rešenja?

Obično se u algoritmima pribegava ispitivanju kratkoročnih posledica svakog od uopšte mogućih poteza tako što se analizira stanje nastalo u „bliskoj budućnosti" kao njegova posledica. Ako se konstatuje da su kola već na početku krenula nizbrdo, tada nema svrhe dalje tračiti vreme nego treba preći na ispitivanja drugih kombinacija. Takav razon nam je uostalom svima blizak iz ličnog iskustva. Niko ko poznaje makar osnove šahovske igre neće uludo poklanjati svoje figure, niti izletati kraljem na sredinu table ili ponavljati stalno iste poteze. A ipak, vrhunski majstori često čine upravo to! Puštaju da protivnik pomisli kako su pogrešili, da bi ga u sledećem trenutku zaskočili nekim neprijatnim iznenađenjem!

Računari ispoljavaju potpunu nemoć pred situacijama gde je privremen gubitak na materijalu zarad jačanja pozicije nadoknadiv tek posle većeg broja poteza. Što je privremeno slabljenje izraženije i što je duži put do njegovog revalorizovanja, to je računaru (doduše, i ži



Situacija koja će da "zaglupi" svaki računar

vom igraču) laza da, kako bi matematičari rekli, preskoči lokalni minimum i prepozna hotimičnu žrtvu u gomili uslinu promašenih poleza

Iz literature smo iskopali divan primer kao ilustraciju za nesnalaženje računara u ovakvim situacijama. Na priloženom dijagramu vidi se pozicija dostignuta nakon desetog poteza u čuvenoj partiji između Emanuela Laskera i njegovog izazivača Tomasa, odigranoj 1912. godine u Londonu. Davnašnji svetski prvak vodeći bele figure već na početku preuzima inicijativu dok se crni štiti Holandskom odbranom, beli je na redu i ima sigurnu pobedu pod uslovom da zna odigrati pravi potez. Lasker je naravno znao – dama jede pešaka na h7 i daje šah crnom kralju! Crni je primoran da prihvati žrtvu ali mu svejedno sledi brzi kraj

11	Dh5 h7+!!	Kg8 h7
12.	Se4 f6++	Kh7 h6
13	Se5-g4+	Kh6-g5
14	h2 h4+	Kg5 f4
15	g2-g3+	Kf4-f3
16	Ld3-e2+	Kf3-g2
17	Th1-h2+	Kg2 g1
18	0-0-0 mat.	

Dati damu za pešaka poslednja je što bi jedan šahovski program uzeo u ozbiljno razmatranje – odnosno prvo što bi odbacio. Stečena poziciona prednost realizuje se najranije za sedam poteza (prema jednom kasnije pronađenom rešenju), koliko daleko nijedan od njih nije ni teoretski u stanju da ide. Ni Chess Master, Battle Chess, Cyrus, Psion ni neki drugi kojem je bio zadavan gornji problem nije uspeo da se „doseti" pobedničkog poteza ma koliko mu sati za razmišljanje bilo stavljano na raspolaganje. Očigledno, još uvek smo veoma daleko od šahovskih programa koji bi se dičili oštroumnošću i nadahnućem jednog Laskera. Sve dok ne počnu razmišljati na isti način kao što to i čovek čini, čovek će ih moći i nadmudriti bez obzira na kako moćnim računarima se vrteli, gruba procesorska snaga sama po sebi nikad neće biti dovoljna da odnese prevagu. A i ako im se ikad budu uspele podariti najdragocenije vrlina ljudskog uma – sposobnost učenja na greškama, maštovitost, duhovitost i prilagodljivost – moraće neizbežno zajedno sa njima preuzeti i sve one dobro znane ljudske slabosti, te ni onde neće biti nepobedivi! U tome možda i jeste lepota šaha

NE SKRAĆUJ, SORTIRAJ!

A da li je izvodljivo značajno ubrzati ispitivanje, a da se ipak unapred ne odbaci nijedna kombinacija? Za neverovati, ali jeste! Da bi smo objasnili kako, prisetimo se jednog pravila dobrog programiranja (koje se nažalost često zaboravlja)

Kada u nekom svom programu pišemo logički *boolean* izraz sastavljen iz više uslova povezanih *or* operandima, nastojimo procenili koji od uslova imaju veće šanse da budu ispunjeni a koji manje, pa ih onda navodimo od najverovatnijeg ka manje verovatnim – sve u cilju da se do rezultata našeg izraza dođe što ranije gledano kroz veliki broj slučajeva. Jer, razvoj višestrukog *or* izraza se prekida čim se dođe do uslova koji je ispunjen, dok se uslovi iza njega ignorišu – kakvi god da su, neće imati uticaja na krajnji ishod. Slično tome, kada se radi o disjunkciji, to jest o više logičkih varijabli vezanih *and* operandima, navode se prvo one koja imaju najmanju verovatnoću da ispadnu tačne, pošto se razvoj izraza obustavlja kod prve netačne

Sa stanovišta računara, neki od poteza koji se razmatraju je valjan ako se posle bilo kojeg protivnikovog dolazi do sigurne pobede ili barem do znatno povoljnije pozicije. Suprotno loš je onaj potez posle kojeg protivnik može da odigra takav svoj koji će ga sigurno dovesti do pobede ili dobijene pozicije. Čim dođemo do poteza koji vodi ka jednoj od pomenutih izvesnosti prekidamo dalje ispitivanje preostalih; ako je u pitanju siguran poraz, vraćamo se korak unazad i tragamo za drugim putem sa povoljnijim ishodištem (ili predajemo partiju ako nameramo gde da se vratimo, tj. ako je posredi stvarna a ne pretpostavljena pozicija); ako je u pitanju sigurna pobeda, tada već znamo šta treba odigrati, te nas preostali potezi ni ne interesuju. E sad, kada sve navedeno uzmemo u obzir, dolazimo do zaključka da bi bilo veoma pogodno poteze koje ćemo razmatrati prethodno sortirati, i to u opadajućem redosledu prema nekoj a priori procenjenoj verovatnoći za pozitivan ishod, dakle tako da prvo dođu oni za koje se pretpostavlja da će najbrže rezultirati uspehom. Ako se naša pretpostavka pokaže tačnom, a statistički posmatrano u najvećem broju slučajeva hoće, sve preostale poteze nećemo morati razmatrati, čime značajno štedimo na vremenu. Vremenska ušteda, što je najzgodnije, istog je reda kao i prirast algoritma – eksponencijalna!

Preliminarna procena može u ovom slučaju biti neprecizna koliko god hoće, ali se to neće odraziti na krajnji rezultat. I kad se dogodi da se neki realno najjači potaz nađe pri kraju liste i kad se na njenom početku nađu potezi za koje će se kasnijom analizom ispostaviti da su bezvredni, ne dolazi do previda već samo do nešto dužeg „razmišljanja". Inače ionako će se ispitati svi potezi ukoliko se ne nađe ni na jedan koji vodi ka radikalno drugačijem odnosu snaga

PET U NIZU

U sledećem nastavku ćemo doneti i odgovarajući programski primer. Naravno, bilo bi previše očekivati od nas da objavimo listing nekakvog programa koji bi igrao šah, ali za neku jednostavniju igricu nema problema. Reč je o implementaciji poznate đačke igre koja je u vreme kada je gorepotpisani išao u srednju školu, predstavljala najpopularnije sredstvo za ubijanje vremena na dosadnim časovima (pošto u razredu nismo imali nijednu devojku). Dovoljne bi bile dve flomastera i parče karo-hartije na koji dvojica igrača onda naizmenično upisuju kružić ili krstić, sve dok jedan od njih ne sastavi u nizu pet svojih znakova horizontalno, vertikalno ili dijagonalno. Ali o tome više u sledećem broju

Nastavak sa strane 77

Nygaard, E. - Taksrud, V
Norveška liga 1993



Crni je igrao u ovoj poziciji 59. Lf2 i jedva dobio kroz 15 poteza. *Chess Genius* je partiju rešio za 6 sekundi! 59. Lf4!! 60 Kf4 Dg2!!! i beli nema odbrane od mata

pozicije jer ona predstavljaju samu suštinu neke pozicije!

Selective Depth nije ono što mu ime kaže! Tu se ne nameštа dubina proračuna kao što to izgleda, jer se maksimalna dubina proračuna ne objavljuje u dokumentaciji. Verovatno je ograničena samo kriterijumima kojima se procenjuje pozicija u krajnjim čvorištima. Znači, program ide toliko duboko dok nije u stanju da jasno proceni poziciju. Ova opcija vam pruža priliku da produžite proračun od datog preseka za 0-12 (12 je po *default*-u) poluhodova!? To je dodatno obezbeđenje jer zamislite da vam se desi da procenite poziciju kao bolju, kad ono protivnik u sledećem potezu forsira mat u 2 poteza!

Pawn Value je još jedna razlika u odnosu na sve ostale programe. Bez detaljne dokumentovanosti je teško je tvrditi kakva je korist od toga. Uglavnom može se umanjiti ili uvećati vrednost svake figure do nekih 25%. Pretpostavljam da ako smanjite vrednost pešaka za 25% to treba da naveda računar da manje ceni pešake i da ih onda lakše žrtvuje, znači da igra oštrije!?

I tu se lista posebnosti „Šahovskog genija" završava. Što je zaista izvrsno! Nema nepotrebne šminke, preostaje samo gola šahovska snaga, koja je zaista pogolema! Da se ne bismo zadržavali u domenu rejtinga, predlažem čitaocima najmerodavniji način testiranja – igranje mečeva sa računarem. Ja sam napravio ogroman napredak u brzopoteznim mečevima za mesec dana, od 1:6 do 4:6 prosečno. I to tek kad sam počeo da igram na način koji mi uopšte ne odgovara – dugo poziciono nadmudrivanje. Bez daljnjeg, ovaj program je najjači od svih koji su se do sad pojavili kod nas!

U OČEKIVANJU ONOG PRAVOG

Sve u svemu, ve[illegible] izbor šahovskih programa na tržištu. I mada izgleda da među njima postoje velika razlika, svi su on[illegible] na nek[illegible] i pate od istih propusta, nedostataka. Ne postoj[illegible] asad ni jedan program koji b[illegible] zadovoljava sve potrebe igrača – uvek nedostaje ponešto što se mora nadoknaditi kod nekog drugog programa. Zato je u ovoj situaciji, bez pravog favorita, najbolje zadržati ih sve dok se stvari ne iskristališu

Valjda će se uskoro pojaviti program koji će vas naterati da obrišete sve ostale sa disketa, te malo podigli kriterijume za programere. Trenutno su najveći hit programi *Fritz2*, *MChess Pro*, *Chess Genius 2.0*, *Chessmaster 3000* i *Zarkov*. O njima, nakon drugom prilikom. Za sada dajemo samo tabelu sa rejtingom programa onako kako su rangirani po nekim časopisima

1	CHESS GENIUS	2495
2	MCHESS PRO	2460
3	SOCRATES	2420
4	MCHESS	2400
5	FRITZ II	2360
6	ZARKOV 2.61	2310

Zoran Kehler

Upravljanje memorijom pod *Windows*-om

Microsoft Windows obezbeđuje grafičku radnu okolinu u kojoj se mogu izvršavati postojeća MS DOS aplikacija i *Windows* aplikacija za koje ne važi DOS-ovo ograničenje memorije na 640kB. *Windows* pozna je tri tipa memorije – konvencionalnu, produženu (*Extended*) i proširenu (*Expanded*), u zavisnosti da li se izvršava u standardnom ili zaštićenom (*Enhanced*) načinu rada. Pored ova tri tipa memorije, PC-kompatibilni računari mogu pristupiti i području visoke memorije veličine 384kB koje se naziva *Upper Memory Area* (UMA).

Prvih 640kB memorije čine konvencionalnu memoriju. MS DOS ima ograničenje da može adresirati samo 1024kB memorije u području konvencionalne i visoke memorije. Kada se pod *Windows*-om izvršavaju DOS aplikacije, ova ograničenja su važna, ali ne i kada se u standardnom i zaštićenom načinu rada izvršavaju *Windows* aplikacije. Ako je *Windows* u zaštićenom načinu rada, moguće je pokrenuti više DOS aplikacija u više virtuelnih DOS mašina. Jedno odstupanje od ograničenja u opsegu memorije u DOS virtuelnoj mašini je kod aplikacija koje su napisane tako da koriste DPMI (*DOS Protected Mode Interface*) specifikaciju. U tom slučaju takvi programi mogu direktno koristiti do 16MB memorije. Primer takvog programa je i novi PKZIP arhivar (verzija 2.xx).

Između vrha konvencionalne memorije (640kB) i početka produžene memorije (1024kB) nalazi se područje visoke memorije (UMA), veličine 384kB. Na memorijske adrese koje leže u opsegu A000-FFFF (640-1024kB) mapirani su sistemski BIOS, ROM memorije na video kartici i eventualnim dodatnim mrežnim i drugim karticama. Ovim smo zapravo rekli da PC kompatibilan računar sa instaliranih 640kB memorije ima adresni prostor veličine 1MB: prvih 640kB čini konvencionalna memorija, a preostalih 384kB je prostor visoke memorije. Mašina sa 1MB fizički instalirane memorije ima adresni prostor od 1408kB: 640kB konvencionalna memorija, 384kB visoke memorije i 384kB produžene memorije koja počinje od 1024kB.

Produžena (*Extended*) memorija je najjednostavniji vid dodatne memorije. *Windows 3.0* i *3.1* koriste produženu memoriju kada rade u standardnom i zaštićenom modu. Na mašinama sa 8088 ili 8086 procesorom nije moguće instalirati produženu memoriju zbog hardverskog ograničenja tih procesora, čiji adresni prostor obuhvata samo 1024kB. Na takvim računarima ne može se izvršavati *Windows*, koji za rad u standardnom modu zahteva bar 256kB produžene memorije. Procesor 80286 može adresirati 16MB memorije, a 80386 može adresirati do 4GB memorije.

Windows 3.x i sve aplikacije pod njim pristupaju produženoj memoriji preko drajvera koji implementiraju XMS specifikaciju (*Microsoft eXtended Memory Specification*). To je najčešće čuveni HIMEM.SYS koji na vaš sistem instalira već i sam MS DOS. Stare aplikacije do produžene memorije dolaze korišćenjem interapta 15, servis 88h; ako takva aplikacija pokuša da to uradi pored instaliranog HIMEM.SYS drajvera, pristup joj neće uspeti.

Proširena (*Expanded*) memorija je još jedan tip dodatne memorije. Sam *Windows 3.x* je ne koristi, ali je može obezbediti MS DOS aplikacijama koje se pod njim izvršavaju. U upotrebi su dve LIM (Lotus/Intel/Microsoft) standarda za proširenu memoriju: 3.2 i 4.0. Pored mogućnosti da se u EMS memoriji drže podaci, za razliku od LIM 3.2, verzija 4.0 dozvoljava da se u proširenoj memoriji izvršavaju i programi, što omogućava multitasking. U MS DOS-u usluga obezbeđivanja EMS memorije za aplikacije pružaju razni programi za upravljanje memorijom, kao što su EMM386.EXE ili QEMM386.SYS.

Windows 3.1 u standardnom načinu rada

Kada otkucate WIN /S i pritisnete Enter, Windows startuje tretirajući konvencionalnu i produženu memoriju kao celinu. Praktično ograničenja za MS DOS aplikacije u ovakvom načinu rada *Windows*-a jesu da moraju koristiti samo konvencionalnu memoriju. Zoni visoke memorije se u ovom slučaju ne pristupa. Takođe, u standardnom modu ne koristi se ni EMS memorija. MS DOS aplikacije je mogu koristiti jedino ako je u mašini instalirana fizički, odnosno na kartici sa EMS memorijom. Produženoj memoriji pristupa se preko HIMEM.SYS drajvera.

Enhanced mode

Pored zbira konvencionalne i XMS memorije *Windows* u zaštićenom načinu rada može koristiti prostor na tvrdom disku kao virtuelnu memoriju, što vi vidite kao tajanstveni i misteriozni „swap file". U ovom načinu rada MS DOS aplikacije se izvršavaju u virtuelnim DOS mašinama, od kojih svaka startuje sa najviše 640kB konvencionalne memorije. Svaka virtuelna mašina nasleđuje okruženje koje je bilo prisutno kada je *Windows* pokrenut. To znači da svaki rezidentni program i drajver uređaja koji vam je trošio dragocenu konvencionalnu memoriju to isto radi i sada: količina tako zauzete memorije je u virtuelnoj DOS mašini uvećana i za prostor koji sistem koristi za svoje potrebe.

Windows u *enhanced* načinu rada koristi visoku memoriju (UMA) na dva načina: za smeštanje translacionih zona MS-DOS programskog interfejsa (MS DOS API) i za smeštanje „prozora" u LIM 3.2 proširenu memoriju (*page frame*).

Zašto je potreban sloj translacije u programskom interfejsu za MS-DOS pod *Windows*-om? Verovatno ste do sada mnogo puta prokleli činjenicu da se *Windows* sam izvršava preko MS DOS-a. Kopija DOS-a u kojoj je pokrenut *Windows* može koristiti samo konvencionalnu memoriju, isto važi i za sve drajvere i mrežni softver pokrenut pre *Windows*-a. Zaštićeni način rada obezbeđuje u UMA memoriji bafere za translaciju mrežnih i MS-DOS API poziva iz zaštićenog u realni način rada (podsetimo se da je MS DOS napisan za realni način rada 80x86 procesora). Pošto se UMA nalazi u prvih 1MB adresnog prostora, može joj se pristupiti u realnom modu iz MS DOS-a. Pomenuti baferi se ovde koriste kao zona preko koje aplikacije pokrenute u zaštićenom modu mogu razmenjivati informacije sa MS DOS-om i mrežnim drajverima. Ako u visokoj memoriji nema dovoljno mesta, translacioni baferi se mogu smestiti u zonu konvencionalne memorije na uštrb slobodne memorije u svakoj od virtuelnih MS DOS mašina.

Windows-u možete eksplicitno narediti gde da smesti *page frame* za EMS i translaciona bafere. Tome služi *ReservePageFrame* direktiva u [386Enh] sekciji SYSTEM.INI datoteke:

ReservePageFrame = True

Ako je vrednost **True**, što je i podrazumevana vrednost, *Windows* će u zaštićenom načinu rada prvo u UMA smestiti *page frame*, pa tek onda translaciona bafere, ako je ostalo dovoljno mesta. Time ste se osigurali da će MS DOS aplikacije pod *Windows*-om imati na raspolaganju EMS memoriju, možda na štetu korišćenja konvencionalne memorije za smeštaj translacionih bafera. Vrednost **False** učiniće da se prvo smeste baferi, pa tek onda *page frame* za EMS, time se obezbeđuje i najviše moguće memorije za MS-DOS virtuelne mašine. Važna napomena: ako se koristi EMM386.EXE, translacioni baferi se uvek smeštaju u konvencionalnu memoriju.

Podršku za EMS memoriju možete potpuno isključiti direktivom **NoEMMDriver** = Yes u [386Enh] sekciji SYSTEM.INI datoteke. Zbog mogućih konflikata u UMA, pojedine zone memorije mogu se učiniti nedostupnim za EMS mapiranje direktivom **EMMExclude** = ; postoji i direktiva sa obrnutim dejstvom: ako *Windows* automatski koristi sve slobodne stranice u UMA za EMS mapiranje, ponekad se pojavi potreba da se neka zona eksplicitno obuhvati mapiranjem, za šta služi direktiva **EMMInclude** = . Desno od znaka jednakosti navodi se opseg adresa u formatu XXXX-YYYY.

Windows u zaštićenom načinu rada kao celinu koristi konvencionalnu, produženu i virtuelnu memoriju. MS DOS aplikacije pod *Windows*-om mogu u virtuelnoj mašini pristupati produženoj memoriji. *Windows* koristi XMS drajver (HIMEM.SYS ili sličan) da bi smestio sebe i svoje drajvere u produženu memoriju prilikom startovanja, zbog toga je za zaštićeni način rada neophodno postojanje produžene memorije.

Virtuelna memorija

Ako virtuelnu memoriju tretiramo kao crnu kutiju, ono što se spolja vidi je da uz pomoć tog koncepta možete istovremeno izvršavati više programa nego što to dozvoljava količina memorije fizički instalirana na računaru. Virtuelna memorija se već par decenija koristi u svetu velikih i srednjih računara, a u svetu personalnih računara relativno je kratko prisutna. U *Windows*-u ćete naći na trag njenog delovanja ako otvorite **Help/About** dijalog *Windows* aplikacija: ako ste u zaštićenom načinu rada, koristite *swap file*, količina memorije koju će prijaviti *Windows* biće veća od fizički instalirane memorije. Aplikacije za *Windows* ne moraju biti specijalno napisane da bi koristile ovu blagodet – one su i u ovom pogledu nezavisne od okruženja (*device independent*) i u potpunosti prepuštaju *Windows*-u da odlučuje na koji će način ispuniti njihove zahteve za dodelom memorije. Naravno, ima i loših strana ovakvog sistema: ako nemate dovoljno fizičke, bićete prisiljeni da zavisite od virtuelne memorije, aplikacija će se izvršavati sporije.

Kako rade sistemi sa virtuelnom memorijom? Dok se izvršava program koji koristi ovakav sistem upravljanja memorijom, u jednom trenutku deo programa i deo podataka nalaze se u fizičkoj memoriji, dok je drugi deo sklonjen na disk u datoteku za straničenje („swapped", odakle i izraz „swap file"). Programi koriste adresni prostor koji je nezavisan od fizičkog memorijskog prostora. Procesor generiše virtuelne ili logičke adrese, mehanizam koji obrađuje ove virtuelne adrese i prevodi ih u adrese stvarnih memorijskih lokacija obično se implementira kombinacijom softverskih i hardverskih komponenti. *Windows* koristi relativno jednostavan oblik translacije virtuelnih adresa u fizičke adrese, koji pretpostavlja da su memorija i programi podeljeni u stranice fiksne dužine. Veličina stranice za *Windows* je 4kB. Programu može biti dodeljena fizička ili virtuelna stranica. Broj fizičkih stranica je jednostavno veličina memorije u kB podeljena sa 4kB.

Kada se pokuša pristupiti adresi koja nije u fizičkoj memoriji, generiše se stranični prekid i kontrolu preuzima VMM (*Virtual Memory Manager*), deo zaštićenog moda *Windows*-a zadužen za upravljanje virtuelnom memorijom. Traženi deo programa ili podataka se učitava u fizičku memoriju sa diska i, ako je potrebno, deo sadržaja fizičke memorije se premešta na disk. Koji će se deo memorije premestiti na disk određuje se LRU (*Least Recently Used*) algoritmom: prvo se izbacuju stranice kojima se najduže nije pristupalo. Ono što vi vidite od celog ovog procesa je treperenje indikatora rada diska i ponekad primetno usporenje rada programa.

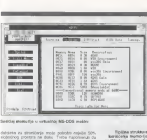
U fizičku memoriju se dovlače samo stranice čiji se sadržaj eksplicitno traži, pa se takav način straničenja naziva „straničenje na zahtev" (*Demand Paging*). Pošto je *Windows* višeprogramski sistem, VMM mora voditi računa i kojem od aktivnih procesa pripada svaka stranica u memoriji ili na disku. VMM formira tabelu stranica, u kojoj svaki ulaz sadrži adresu stranice u memoriji ili pointer (ili drugu informaciju) na položaj stranice u sekundarnoj memoriji, kao i više kontrolnih bita, među kojima su indikator da li je stranici pristupano i da li je „prljava", odnosno da li se u nju nešto pisalo. Svaki upis postavlja oba indikatora u aktivno stanje jer se pisanje tretira kao pristup stranici.

Swap datoteke

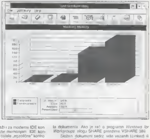
Zaštićeni način rada sistema *Windows 3.x* može da koristi privremenu ili trajnu *swap* datoteku, odnosno datoteku za straničenje. Pre nego što požurite i pokušate da datoteku za straničenje smestite na RAM disk, upozoriću vas da je to u korist vaše štete: da biste formirali RAM disk, morate *Windows*-u oduzeti deo fizičke memorije, u taj deo fizičke memorije smeštate virtuelnu memoriju koja vam je potrebna zato što ste iskoristili fizičku memoriju da biste u nju smestili RAM disk.

Windows zahteva najmanje 512kB slobodnog prostora na disku za privremenu datoteku za straničenje. To je obična MS-DOS datoteka imenom WIN386.SWP koja se po potrebi smanjuje i povećava. Ona se automatski briše kada se iz *Windows*-a izađe na regularan način. Najveća veličina privremene *swap* datoteke može se zadati direktivom **MaxPagingFileSize** = u SYSTEM.INI datoteci; sa druge strane, ona je ograničena eventualnom **MinUserDiskSpace** = direktivom kojom se zadaje koliko se slobodnog prostora mora ostaviti na disku u bilo kojem trenutku. Ova se datoteka kreira na mestu koje je određeno direktivom **PagingFile**.

Veličina privremene *swap* datoteke je veličina operativne memorije pomnožena sa četiri, zaokruženo na sledeći umnožak od 4MB. Na primer, ako imate 3MB slobodne XMS memorije, veličina *swap* datoteke biće 16MB. Zapravo, ne mora se uvek množiti sa 4, množi se sa vrednošću zadatom direktivom **PageOverCommit** = u SYSTEM.INI datoteci, a 4 je podrazumevana vrednost. Preporučena vrednost je ograničena i pravilom „pola slobodnog prostora", odnosno da



Sadržaj memorije u virtuelnoj MS-DOS mašini



Tipična struktura korišćenja memorije za Windows aplikacije

datoteka za straničenje može potrošiti najviše 50% slobodnog prostora na disku. Treba napomenuti da se komprimovani prostor na disku ne smatra upotrebljivim.

Trajna datoteka za straničenje zauzima uzastopne lokacije na tvrdom disku i kreira se u korenom direktorijumu jednog od instaliranih tvrdih diskova. Brže joj se pristupa nego standardnoj MS-DOS datoteci, pa je i rad sa njom brži nego sa privremenom datotekom – zbog toga što svaki pristup datoteci kroz MS DOS zahteva prebacivanje procesora iz zaštićenog (32-bitnog) u realni (16-bitni) način rada, jer je MS DOS napisan za realni način rada 80x86 procesora. Trajna datoteka za straničenje nosi ime 386SPART.PAR i ima atribute skrivene sistemske datoteke, pa je u običnom listingu sadržaja direktorijuma ne vidite, niti je možete obrisati bez dodatnog posla. Pošto mora biti locirana u uzastopnim lokacijama na disku, njena najveća dužina ograničena je veličinom najvećeg komada slobodnog prostora na disku.

Trajna datoteka za straničenje se kreira iz **Control Panel/386 Enhanced/Virtual Memory** dijaloga. *Windows* tada na disku pravi SPART.PAR datoteku čiji sadržaj ukazuje na položaj i veličinu *swap* datoteke. Ako obrišete SPART.PAR, *Windows* neće moći da nađe datoteku za straničenje, niti će moći da je obriše ili izmeni njenu veličinu ili položaj. Možda ste već doživeli nesnalaženje *Windows*-a u zaštićenom načinu rada kada ste instalaciju *Windows*-a kopirali sa jednog diska ili računara na drugi, jer se kopiranjem obično ne prenesu skrivene i sistemske datoteke, pa je *Windows* ostao bez jedne od komponenata svog sistema za virtuelnu memoriju.

Trajnu datoteku za straničenje *Windows* može kreirati samo na tvrdim diskovima koji podržavaju standardni Int 13h interfejs. Odmah nakon pojave *Windows 3.1* stvorila se tema da SCSI diskovi i adapteri ne spadaju u tu kategoriju, što nije tačno – najveći deo SCSI disk sistema podržava taj tip interfejsa. VMM podržava samo diskove koji za *swap* datoteku koriste sektore veličine 512 bajta. Sektore drugih veličina koriste razni programi za particionisanje diska i VMM odbija saradnju sa njima, uz izuzetak Compaqovog ENHDISK.SYS drajvera.

Ako u **Virtual Memory** dijalogu **Control Panel**-a otkrijete da vam *Windows* nudi *swap* datoteku manju od slobodnog prostora na disku, znači da je slobodan prostor fragmentiran, odnosno da ga ne čine uzastopne lokacije na disku. Problem ćete otkloniti korišćenjem programa za defragmentaciju diska, kao što su *Speed Disk* iz *Norton Utilities*, *Compress* iz *PC Tools*-a, *Defrag* iz MS DOS-a 6.x ili drugi. Ako ovo otkrijete kada budete želeli da povećate već postojeću datoteku za straničenje, moraćete prvo obrisati staru (iz **Control Panel**-a – ne zaboravite!), defragmentirati disk, pa zatim kreirati novu, veću datoteku.

Korišćenjem drajvera za 32-bitni pristup disku (*32-bit block device driver*) *Windows* će imati više koristi od *swap* datoteke. Straničenje je u tom slučaju brže i MS DOS programi se mogu premeštati na disk i kada su aktivni. *Windows* se isporučuje samo sa drajverom za AT BUS kontrolere tvrdog diska zasnovane na Western Digital WD 1003 IDE kontroleru. U SYSTEM.INI datoteci naći ćete ga kao **device = WDCTRL**. Kako SCSI diskovi i kontroleri ne spadaju u ovu odabranu grupu, većini SCSI diskova ne može se pristupati 32-bitno. Isto važi i za moderne IDE kontrolere sa hardverskom *cache* memorijom, IDE kontrolere za VESA Local Bus i ostale „egzotične" kontrolere. Očekuje se da proizvođači ovakvog hardvera stave korisnicima na raspolaganje drajvere za 32-bitni pristup disku (VxD drajvere), ali to ide dosta sporo.

Straničenje na mreži

Korišćenje mrežnog diska za straničenje se nikako ne može preporučiti, ali ćete možda biti primorani da ga koristite ako imate radnu stanicu bez lokalnog diska. Efektivna brzina prenosa podataka prema mrežnom disku uvek je par puta sporija od brzine prenosa lokalnog diska. Straničenje je samo po sebi proces koji usporava rad *Windows* aplikacija, pa korišćenje mrežnog diska u te svrhe svakako donosi daljnju degradaciju performansi.

No, dobro. Ako već morate tome da pribegnete, primorani ste da koristite privremenu datoteku za straničenje. Direktiva **SwapFile** mora da ukazuje na direktorijum na mrežnom disku koji ne sme biti otvoren samo za čitanje, već na njemu morate imati pravo čitanja i pisanja. Svaki korisnik mora imati svoju datoteku, jer se inače ne bi znalo kome u datom trenutku njen sadržaj pripada.

Moram da priznam da straničenje na mrežni disk spada u „zonu sumraka". Nikada nisam bio primoran da ga koristim, već sam to radio namerno, u eksperimentalne svrhe. Pitanje je da li se u realnom poslovnom okruženju pojavljuje potreba za time, posebno zbog toga što će *Windows* raditi i bez virtuelne memorije, ili će prodavac ubediti kupca da radna stanica na kojoj će se *Windows* koristiti mora imati disk.

Budimo u toku: zašto *Word for Windows 6.0* zahteva SHARE?

Ako ste u *Windowsu 3.1* probali da instalirate novi *Word for Windows 6.0*, možda ste se susreli sa problemima oko SHARE.EXE programa. *Word*-ov *Setup* ga instalira, a ako ga vi uklonite, *Word* ne radi pravilno.

Verovatno ste smatrali da je SHARE.EXE potreban samo za rad na mreži, radi kontrole pristupa više korisnika ili aplikacija jednoj datoteci. Ovde njegova primena nema veze sa mrežnim radom, već sa OLE 2.0 sposobnostima ugrađenim u *Word*. Razlog za instalaciju SHARE programa leži u tome što implementacija OLE 2.0 ugrađena u *Word 6.0* koristi koncept složenih dokumenata (*Compound Documents*), koji opet zahteva SHARE zbog njegovih sposobnosti kontrole pristupa datotekama i to zaključavanja samo dela dokumenta. Ako je reč o programu *Windows for Workgroups*, ulogu SHARE preuzima VSHARE.386.

Složen dokument sadrži više vezanih (*Linked*) ili ugrađenih (*Embedded*) objekata iz drugih aplikacija ili dokumenata i zapravo je osnova za OLE koncept. Zaključavanje dela dokumenta (*range locking*) dozvoljava aplikaciji koja je OLE server da radi editovanja objekta zaključa samo deo dokumenta koji sadrži objekat.

Iako aplikacije koje podržavaju OLE 2.0 ne moraju da koriste složene datoteke, njihova upotreba se preporučuje jer sistem datoteka standarda OLE 2.0 obezbeđuje efikasan i fleksibilan pristup podacima u objektima. On omogućava da se objekti ili delovi objekata učitavaju sa diska u memoriju bez potrebe da se u memoriju učita ceo dokument.

Implementaciju OLE 2.0 složenih dokumenata realizuje biblioteka STORAGE.DLL. *Word* i druge OLE 2.0 aplikacije mogu pisati u STORAGE.DLL, koja će onda promene preneti u datoteke na disku. STORAGE.DLL zahteva funkcije zaključavanja bela datoteka koje se u *Windows*-u omogućavaju na način prikazan u tabeli.

	Enhanced Mode	Standard Mode	Isporučuje se sa
Windows 3.1	SHARE.EXE	SHARE.EXE	name
Windows for Workgroups	SHARE.EXE i VSHARE.EXE	SHARE.EXE	VSHARE.EXE

Microsoft preporučuje da se SHARE koristi sa sledećim parametrima:

SHARE /L:500 /F:1500

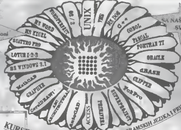
„L" se odnosi na broj lokota, „F" na prostor predviđen za podatke o datotekama. Svaki lokot zahteva 19 bajta. *Word* zahteva najmanje tri lokota za svaku datoteku koju otvorite.

Šta vam Microsoft nije rekao o *TrueType* fontovima

Mnogi od problema sa štampanjem pod *Windows*-om su nestali aprila 1992. godine, kada je Microsoft u novi *Windows 3.1* ugradio skalabilne *TrueType* fontove. Setite se samo muke oko instalacije ekranskih i štampačkih fontova sa našim latiničnim znacima u davnim danima *Windows 3.0*. Doduše, te muke donele su i novu sortu specijalista, stručnjake za razne pakete za generisanje i rukovanje fontovima i prave male manufakture koje su od Bitstream, Agfa Compugraphics i drugih komercijalnih paketa fontova „proizvodili" naše latinične i ćirilične verzije.

Microsoft kao da je bio pomalo stidljiv u promociji svih dobrih osobina *TrueType* fontova – osobina koje bi se teško mogla očekivati od firme koja za svaki novi korak u razvoju softvera, novu mogućnost neke od aplikacija ili jednostavno novi izgled korisničkog interfejsa tvrdi da je „ono pravo", revolucionarna promena, i obično naziva bombastičnim imenom. Ono što nam dakle Microsoft nije rekao jeste da *TrueType* fontovi imaju osobinu da se mogu ugraditi (*Embedding*) u dokument, tako da se dokument može odštampati u punom sjaju i na računaru na kome

Nastavak na strani 118


UNIX
VMS
MS DOS
C++
COBOL
PASCAL
FORTRAN 77
ORACLE
dBASE
CLIPPER
FoxPro
SUPERPROJECT
PRIMAVERA
ACCESS
AutoCAD
ARCHITECTURAL
CorelDRAW!
GRAPHER
MathCAD
MS WINDOWS 3.1
LOTUS 1-2-3
QUATTRO PRO
MS EXCEL
MS WORD
WORDPERFECT
OS / 2

BAZEN KOJI SE VEČNO PUNI

Iako je zvučala sasvim jednostavno 91. pitalica je zapravo bila problem koji je zahtevao više matematičkog znanja nego ijedan do sada postavljeni zadatak. Srećom, naši čitaoci se nisu uplašili i bili su nagrađeni prilično neočekivanim rezultatom.

Podsetimo se pre svega postavke. Na slici 1 prikazan je bazen koji se puni kroz jednu cev a prazni kroz drugu. Ako je prva slavina otvorena a druga zatvorena, prazan bazen će se napuniti za 5 časova. Ako je prva slavina zatvorena a druga otvorena, pun bazen će se isprazniti za 10 časova. Pitali smo za koliko će se vremena napuniti prazan bazen ako su otvorene obe slavine.

Problemi nalik na ovaj česti su u raznim zbirkama zadataka iz rekreativne matematike i obično se rešavaju otprilike ovako: prva slavina za jedan sat puni petinu bazena, a za isto vreme druga slavina isprazni desetinu bazena. U zbiru, za sat se napuni 1/5–1/10 = 1/10 bazena, dakle čitav bazen će se napuniti za 10 sati. U ovom rešenju sporna je druga slavina – da bi proces tekao na taj način, trebalo bi da se voda istače nekom pumpom koja će u jedinici vremena izvući konstantnu količinu. U realnosti brzina isticanja vode zavisi od njenog nivoa – kada je bazen pun voda će isticati brzo, a kada je bazen gotovo prazan isticanje je znatno sporije. Zapravo, ako se uzme da je površina bazena mnogo veća od površine poprečnog preseka cevi i zanemari razlika u vazdušnom pritisku na nivou vrha i na nivou dna bazena (obe pretpostavke su sasvim opravdane), dobija se da je V_i=sqrt(2gh) gde je V brzina isticanja vode, h trenutna visina vode u bazenu, a g gravitaciono ubrzanje $g = 9.81\ m/s^2$.

Posmatrajmo sada proces pražnjenja bazena, tj. situaciju u kojoj je slavina 1 zatvorena a slavina 2 otvorena. Ako sa P obeležimo površinu poprečnog preseka bazena, sa S površinu poprečnog preseka cevi, a sa V brzinu isticanja vode, važiće relacija (1) sa slike 2. Znak minus potiče otuda što se bazen prazni tj. nivo opada. Ako integralimo levu i desnu stranu i primenimo granične uslove (u početnom trenutku visina je H a vreme 0, dok je u završnom trenutku visina 0 sa vreme t_2 = 10 časova – jednačina (2) sa iste slike) dobija se relacija (3). Iz nje ne možemo da izračunamo visinu bazena jer nam nisu poznata dva poprečna preseka, ali treba da primetimo da su ti preseci konstante i da postoji solidna nada da će se pojaviti i u završnoj jednačini – dakle, dobili smo visinu bazena u zavisnosti od datog vremena t_2 i nekih drugih konstanti.

Prelazimo sada na traženi slučaj u kome su obe slavine otvorene. Ako brzinu protok vode kroz prvu slavinu označimo sa Q (znamo da je t_1 = P*H/Q = 5 časova) dobićemo P*dh=(Q-S*V)*dt odnosno, po grupisanju promenljivih i zameni V, izraz (4). Da vidimo kakvi su početni uslovi. U trenutku t = 0 brzina isticanja je 0 jer je bazen prazan pa je Q>SV, bazen počinje da se puni. Samim tim brzina isticanja vode počinje da raste da bi u nekom trenutku postala jednaka brzini doticanja vode: Q=S*sqrt(2gH_s) gde smo sa H_s obeležili ravnotežnu visinu. Radi lakšeg pisanja uvešćemo veličinu $T=t_2/t_1$ i dobiti relaciju (5). Za $H=H_s$ biće očito T=2, što znači da je u našem slučaju ($t_2=2t_1$=10 časova) željena visina bazena upravo jednaka ravnotežnoj visini.

Da vidimo, dakle, kako će se bazen puniti – odgovor daje jednačina (6) koja se može integraliti uz odgovarajuće granične uslove (7) i tako dobiti traženo vreme punjenja t_p (8). U našem slučaju parametri su „tempirani" tako da jednačina (8) u tački $T=t_2/t_1=2$ ima singularitet, što znači da će se bazen beskonačno puniti i težiti traženoj visini, ali je nikada neće dostići. Da bi se ipak stekao neki pregled o približnom vremenu punjenja bazena, pretpostavićemo da se nivo od 99% traženog može proglasiti „skoro punim" – vreme je tada oko 43 časa. Za 99.99% pun bazen potrebno je čak 89 časova. Pokazuje se da pomenuto rešenje iz raznih zbirki, 10 časova, nije čak ni dobra aproksimacija – za to vreme bazen bi se napunio tek na oko 70% željene visine.

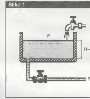
Čitaoci koji su pokušavali da zadatak reše numerički upali su u predviđenu „zamku" kod svih numeričkih integraljenja ili rešavanja jednačina (pa i diferencijalnih) postupak se nastavlja dok se ne dostigne željena tačnost. Recimo da je usvojena tačnost 0.0001 – program će se izvršavati sve dok ne ustanovi da je bazen 99.99% pun i dati rezultat „89 časova". Da je zahtevana veća tačnost, dobio bi se veći broj (i obratno), ali se nikada ne bi dobilo pravo rešenje: bazen se nikada neće napuniti, premda će težiti da bude pun. Isto tako se numeričko rešenje teži da bude tačno, ali tačno neće nikada biti. To je nauk da se numeričke metode ne smeju nekritički primenjivati pre nego što se razmotre suština i matematički model simuliranog procesa.

U predviđenom roku (a i posle njega, pošto su „Računari" kasnili) primili smo 15 tačnih rešenja i čak 57 pogrešnih. Prva nagrada (30 din) je pripala autoru najkompletnijeg rešenja **Petru Beličevu**, dok su preostale dve izvučene: najviše sreće imali su **Vinko Marinković** (20 din) i **Miloš Đermanović** (10 din). Sve trojica nagrađenih su iz Beograda. Zanimljiva rešenja poslali su *Filip Crnogorac* i *Vladimir Nešić*.

Nagrađeni – 91. Pitalica

1	Petar Beličev, Beograd	30 dinara
2	Vinko Marinković, Beograd	20 dinara
3	Miloš Đermanović, Beograd	10 dinara

Slika 1



Bazen koji se puni kroz jednu cev a prazni kroz drugu

Problem iz prvih „Računara"

Sto! broj našeg časopisa je dobra prilika da se podsetimo logičkog problema koji je postavljen još nusre 1984. godine u izdanju „Računari u vašoj kući". Tada, dakako, „Dejanove Pitalice" nisu postojale kao rubrika, ali sve vreme taj zadatak smatramo za prvu Pitalicu. Sticajem okolnosti njegovo rešenje nikada nije objavljeno u „Računarima" pa je ovo možda dobra prilika za svojevrsnu „reprizu".

Napišite program na bejziku koji lista samog sebe! Komanda RUN, dakle, treba da da isti (potpuno identičan, sa svim eventualnim blankovima ispred brojeva linija itd) rezultat kao i komanda LIST. Za testiranje treba koristiti GWBASIC ili neki sa njim kompatibilan interpreter. U programu nisu dopuštene naredbe LIST, PEEK, POKE, pozivi mašinskih potprograma i uopšte sve vrste „trikova" – problem je logičke a ne „hakerske" prirode, dakle nije ideja da na neki volšeban način naterate program da se izlista, nego da veštom manipulacijom sa stringovima proizvedete odgovarajući rezultat.

Programe ćemo rangirati prema dužini – nagradu dobija najkraći program koji ispunjava navedene uslove. Dužinu dobijate tako što otkucate SAVE „ime",A i, pošto izađete u DOS, DIR IME.BAS. Ne zaboravite ono ,A jer se meri dužina ASCII verzije programa a ne njegovog tokenizovanog ekvivalenta.

KAKO DOSTAVITI REŠENJE

Rešenja Pitalice, zajedno sa obrazloženim programima koje ste napisali, pošaljite na uobičajenu adresu „Računari" (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd, tako da prispiju pre 20. maja 1994. godine. Sva pisma sa korektnim rešenjima konkurišu za novčane nagrade u iznosu 30, 20 odnosno 10 dinara, dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača Pitalica. Identifikacioni broj dobijate tako što u prva tri kućice upišete poslednje tri cifre našeg broja telefona, u sledeće dve godinu vašeg rođenja i na kraj dodate dve cifre po izboru. Obratite pažnju da sva rešenja koja šaljete u toku godine nose isti identifikacioni broj. Osim običnom, rešenje ove Pitalice možete da pošaljete i elektronskom poštom posredstvom Sezama – koristite meniju REDAKCIJA, PITALICE, REŠENJE ili iz komandnog moda pošaljite ličnu poruku korisniku pitalice (*mail write pitalice*).

93: Program koji lista samog sebe dug je ______ bajta

Ime i prezime ______

Adresa ______

Mesto ______

Kompjuter ______ Vreme ______

Identifikacioni broj: ⌴⌴⌴ (iz broja tel) ⌴⌴ (god. rođ.) ⌴⌴ (po izboru)

Slika 2

$$\frac{dh}{\sqrt{h}} = -\frac{S}{P}\sqrt{2g}\,dt \qquad (1)$$

$$\int_H^0 \frac{dh}{\sqrt{h}} = -\int_0^{t_2} \frac{S}{P}\sqrt{2g}\,dt \qquad (2)$$

$$t_2 = \frac{P}{S}\sqrt{\frac{2H}{g}} \qquad (3)$$

$$\frac{dh}{dt} = \frac{Q - S\sqrt{2gh}}{P} \qquad (4)$$

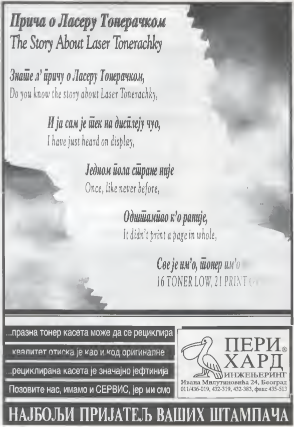
$$T(H) = \frac{Q}{S}\sqrt{\frac{2}{gH}} \qquad (5)$$

$$\frac{dh}{dt} = \frac{H_s}{t_1} - \frac{2\sqrt{H_s}}{t_2}\sqrt{h} \qquad (6)$$

$$\int_0^H \frac{dh}{\frac{H_s}{t_1} - \frac{2\sqrt{H_s}}{t_2}\sqrt{h}} \qquad (7)$$

$$t_p = t_2\left(\frac{T}{2}\ln\frac{T}{T-2} - 1\right) \qquad (8)$$

Relacije kojim se dobija približno vreme punjenja bazena


Прича о Ласеру Тонерачком
The Story About Laser Tonerachky
Знаше л' йричу о Ласеру Тонерачком,
Do you know the story about Laser Tonerachky,
И ја сам је шек на дисйлеју чуо,
I have just heard on display,
Једном йола сшране није
Once, like never before,
Одшшамйао к'о раније,
It didn't print a page in whole,
Све је им'о, шонер им'о
16 TONER LOW, 21 PRINT
...празна тонер касета може да се рециклира
...квалитет отиска је као и код оригиналне
...рециклирана касета је значајно јефтинија
Позовите нас, имамо и СЕРВИС, јер ми смо
ПЕРИ ХАРД ®
ИНЖЕЊЕРИНГ
Ивана Милутиновића 24, Београд
011/436-019, 432-319, 432-383, факс 435-513
НАЈБОЉИ ПРИЈАТЕЉ ВАШИХ ШТАМПАЧА

Zoran Životić

Postoji str funkcija i za to...

Rad sa stringovima je često za početnike zbunjujući posebno ako na C prelaze sa bejzika ili paskala. S obzirom da C ne poznaje tip *string*, rad se svodi na primenu različitih funkcija koje manipulišu nizom karaktera, pa je i za tako trivijalnu operaciju kao što je dodela vrednosti jednog stringa drugom potrebno koristiti funkciju

strcpy(string1, string2),

Da zabuna bude potpuna, postarali su se autori C-a intenzivnom primenom zloglasnih pointera, tako da i dodela

string1 = string2,

ima svoje značenje, ali ono je sasvim drugačije od prethodnog slučaja. Ipak, o pointerima i stringovima sam već dosta pisao u ovoj seriji, pa se nadam da pažljivi čitaoci više nemaju problema sa razumevanjem ovog segmenta C jezika.

Ipak, ono što je i kod iskusnijih programera interesantno je da prvih nekoliko funkcija za rad sa stringovima koje se upoznaju na početku, poput **strcpy, strlen, strcmp**, ponekad **strlwr** i **strupr** često ostaju i jedini fond koji programer koristi. Biblioteke savremenih kompajlera po pravilu sadrže bogat skup funkcija za rad sa stringovima i šteta je ne poznati bar neke od njih, jer efikasno rešavaju česte probleme na koje programer nailazi. Da bih vas zainteresovao da malo prelistate dokumentaciju vašeg kompajlera, evo nekoliko karakterističnih primera.

U velikom broju programa ste sigurno sreli situaciju u kojoj, kada izaberete naziv datoteke, program sam dodaje produžetak naziva. Tako će recimo tekst kome date naziv **text** na disk biti snimljen kao **text.wri** ili **text.doc** ili **text.txt**, u zavisnosti o kom tekst procesoru je reč. Sigurno ste primetili i da ukoliko sami upišete produžetak naziva, program poštuje vašu želju i naziv ne menja. Algoritam je očigledno jednostavan – ako u nazivu koji korisnik otkuca ne postoji znak tačka, program može da doda podrazumevanu ekstenziju, inače ne treba da preduzima ništa. Bez poznavanja **str** bibliotečkih funkcija, programer može da krene sledećim putem

```
i = 0,
while ( filename[i] != 0 && filename[i] != '.' )
      ++i
if ( filename[i] == 0 )
      strcat( filename, ".txt"),
```

Nije strašno, jedna promenljiva i četiri linije koda. Biblioteka ipak nudi mnogo kraće i čitkije rešenje

```
if ( strchr( filename, '.' ) == NULL )
      strcat( filename, ".txt" ),
```

Funkcija **strchr** vraća pointer na prvu pojavu znaka – ukoliko znak ne postoji u stringu, vraća NULL, što je ovde jedino i iskorišćeno. Primer, naravno, važi samo u slučaju kada korisnik ne može da kuca punu stazu do datoteke – ako mu je to omogućeno, treba biti obazriv, jer će i naziv "..\pera" ostati bez produžetka iako to nije zasluži. Ali, i za to ima dobrih **str** funkcija.

Kod bilo kakve analize sadržaja stringa često se javlja potreba da se u njemu pronađe karakteristična sekvenca znakova. Da li se u stringu nalazi reč "demo"? Način na koji se može dobiti efikasan odgovor na ovo pitanje u mnogome zavisi od konkretne situacije. Ukoliko je string koji se pretražuje dugačak (recimo da se radi o tekstu u tekst-procesoru) dolazimo do stalno aktuelnog pitanja algoritma za pretraživanje teksta. Ipak, za svakodnevne situacije u kojima analiziramo kraće stringove, jednostavno rešenje nudi funkcija

```
if ( strstr( string, "demo" ) != NULL )
puts("Rec DEMO se nalazi u stringu")
```

Strstr vraća pointer na prvu pojavu stringa "demo" u stringu **string**. Ukoliko ne postoji, vraća NULL pointer. Ostale pojave reči u istom stringu se lako mogu pronaći na sledeći način

```
char *found,

found = string,
while ( ( found = strstr( found, "demo" ) ) != NULL )
      { // uradi nesto sa tim demoom
        found += strlen( "demo"), }
```

Sličnu situaciju imamo kada želimo da proverimo da li se bilo gde u stringu nalazi neki od znakova iz skupa koji zadajemo drugim stringom. Da ostanemo na terenu datoteka – da li je korisnik u imenu datoteke zadao neki od džoker znakova ? ili *. Umesto dva poziva funkcije **strchr** koju ste već videli, jednostavnije je

```
if ( strpbrk( filename, "?*" ) != NULL )
      // ima dzokera
```

Funkcija **strpbrk** vraća pointer unutar **filename** na prvu pojavu bilo kog od znakova iz zadatog stringa. U ovom primeru se, kao i u prethodnim, funkcija koristi samo za ispitivanje, ali uz jedan pointer koji bi preuzeo rezultat moguće je nastaviti sa obradom. Na primer

```
char *pos,

pos = strpbrk( filename, "?*" ),
if ( pos != NULL )
      printf("Prvi dzoker na poziciji %d\n", pos-filename )
```

"Najkomplikovanija" situacija nastaje kada treba iz stringa redom izdvajati reči – radnje opisane lepim domaćim glagolom – tokenizacija. Iz konfiguracione datoteke čitate liniju i sada je treba rastaviti na sastavne delove, pri čemu razmak nije jedini koji znači kraj/početak reči. Tipično se tu pojavljuju znaci poput tabulatora, navodnici, zarezi i slično, a pri tome se lista često menja od reči do reči u zavisnosti od konkretnog sadržaja linije. Na primer, iz linije

output com2 9600,n,8,1

treba pokupiti prvo reč koja opisuje o kojem parametru se radi, a zatim i same vrednosti. Iako se ovaj posao može manje-više generalno rešiti i primenom **fscanf** ili **sscanf** funkcija, opštije rešenje od toga je

```
char *token,
char *device,

if ( (token = strtok( line, " \t" )) != NULL ) {
      if ( !stricmp( token, "output" ) ) {
            device = strtok( NULL, " \t")
            speed = atoi( strtok( NULL, " ,\t") ).
itd
```

U primeru, naravno, nedostaju provere da li parametar zaista postoji (**strtok** će vratiti NULL kada ostane bez tokena), ali se one lako mogu ugraditi.

Karakteristično za **strtok** funkciju je da se nakon prvog poziva, kada treba proslediti string koji se analizira, za sve dalje pozive navodi NULL, što znači da se nastavlja sa izdvajanjem iz ranije prosleđenog stringa. Druga karakteristika o kojoj treba voditi računa je da se osnovni string menja – u njega funkcija na kraju svakog tokena mora da upiše nulu kako bi izdvojeni token bio propisno terminisan.

Među **str** funkcijama su i neke zaista retko potrebne. Na primer, funkcija

strrev(char *str),

zamenjuje redosled znakova u stringu, tako da prvi znak postaje poslednji itd. Iako funkcija **strdup** može izgledati podjednako nekorisna, ona se može lepo upotrebiti. Funkcija **strdup**, koristeći **malloc** funkciju, odvaja prostor i u njega kopira sadržaj prosleđenog stringa. Efikasno rešenje kada treba vršiti menjanje stringa ne dirajući original, na primer

```
char *prgpath,
char *path,

prgpath = strdup( argv[0] ),
path = strrchr( prgpath, '\\' )
*(path+1) = 0,
```

Ovaj segment koda uzima naziv samog programa (argv[0]) i zatim pronalazi prvu pojavu znaka '\\' od kraja stringa (funkcija **strrchr**) i jedno mesto iza njega terminiše string. Nakon ove operacije pointer **prgpath** pokazuje na stazu do programa. Sada se ovoj stazi možemo dodati naziv neke konfiguracione datoteke, čime obezbeđujemo da se program može startovati iz bilo kog direktorijuma a da konfiguracione datoteke budu locirane na istom mestu na kome i sam program. Za manje iskusne i upozorenje – nije moguće samo

strcat(prgpath, "data.cfg"),

jer **prgpath** ukazuje na dinamički odvojen prostor čija je dužina ograničena originalnim nazivom programa – ako je naziv **cfg** datoteke duži, eto izvora "misterioznih" krahova programa. Dakle, prethodno **strcpy(cfgfile, prgpath)** pa tek onda **strcat**, pri čemu je **cfgfile** novi string dovoljne dužine.

Str funkcije su veoma pažljivo optimizovane, tako da se retko može na brzinu napisati nešto bolje. Odličan primer za to je pomenuta funkcija **strpbrk**. Na "prvu loptu" deluje da je dovoljan običan algoritam tipa "gruba sila" – redom uzimo znak po znak iz prvog stringa i zatim ga poredimo redom sa svakim znakom drugog stringa. Par linija koda, ali izuzetno neefikasno, jer u najgorem slučaju imamo **strlen(string1)*strlen(string2)** poređenja i još nešto više inkrementiranja pointera. Bibliotečka funkcija MSC-a to radi daleko pametnije

```
char *strpbrk( char *string, char *del )
{
      char set[256],
      char c,

      memset( set, 0, sizeof(set) ),
      while ( ( c = *del++ ) != 0 )
            set[ c ] = 1,

      while ( ( c = *string ) != 0 )
            if ( set[ c ] )
                  return string,
            else
                  ++string,
}
```

Dakle, kreira se niz od 256 elemenata i na 1 postavi vrednost onih elemenata čiji se kodovi nalaze u stringu **del**. Zatim je dovoljan samo jedan prolaz kroz string da bi se utvrdilo da li se neki od njegovih znakova nalazio u stringu **del**. Ovde je preglednosti radi odvojen niz od 256 **char** elemenata – funkcija to radi tako što na steku odvoji niz od 256 bitova (32 bajta) i zatim postavlja bit po bit po istom principu. S obzirom da su u ovoj igri samo operacije deljenja sa 8 i pomeranje bitova koje mikroprocesor izvodi efikasno, postignuta je ušteda u prostoru bez posebnog žrtvovanja brzine. Ukoliko nemate prethodnih iskustava sa paskalom, ovakvo rešenje vam verovatno ne bi odmah palo na pamet!

Compile for 286, 386, 486

Programeri željni brzine sa čežnjom često bacaju pogled na opcije kompajlera kojim se "generiše 286 ili 386 ili 486 specifičan set instrukcija" (kod MSC prekidač G0 1 2 3 4 gde je 0=8086, 2=80286 itd). Ima li smisla – da li će DOS program zaista "poleteti" ako kompajleru naložimo da koristi specifičnosti 486 mikroprocesora? Da li dobitak vredi gubitka mogućnosti da program radi i na skromnijem računaru?

Odgovor je, na žalost, kao inače kada se radi o "krpljenju" ograničenja koja nameće zastareli osnovni dizajn računara i operativnog sistema – praktično ne. Te "specifičnosti" koje je kompajler u stanju da iskoristi svode se na veoma, veoma uzak krug instrukcija. Nema praktično nikakvog diranja u osnovni kod programa, bibliotečke funkcije su predviđene i za rad na 8086 procesoru, pa ostaje da se samo u nekim sitnim detaljima postignu određena poboljšanja.

Najčešće poboljšanje je mogućnost 286 da na stek direktno gurne operand instrukcije. Ako pozivate funkciju sa konstantnim parametrom, na primer

test(1)

onda umesto MOV AX,1 PUSH AX dobijate nešto brže i kraće PUSH 1. Isto se može primeniti i u slučajevima kada je parametar adresa statičkog stringa, jer kompajler ovu adresu vidi kao konstantu, pa umesto MOV AX, OFFSET XXX PUSH AX može direktno PUSH OFFSET XXX. Ovo je posebno zgodno kod **far** modela kada je niz **near** tipa, jer se za smeštanje adrese niza na stek može napisati gotovo dvostruko kraći niz instrukcija. Dobici u brzini retko mogu imati ikakvog značaja, dok su dobici u dužini programa nešto značajniji. Kao orijentacija može poslužiti praktičan primer u kome se izvršna datoteka od 162 kb nakon prevođenja opcijom /G2 smanjuje na 156 kb. Ovo naravno treba shvatiti kao veoma nesiguran procenu, jer potpuno zavisi od toga kakav je program.

U MSC verziji 6.0 dodato je interesantno poboljšanje kada se program prevodi za ≥386 mikroprocesore. S obzirom da su operacije sa far adresama veoma česte u **large** modelu, kompajler generiše kod koji koristi 32-bitne verzije registara, pa se lako umesto uobičajenog korišćenja dva registra za segmentni i ofsetni deo adrese, na primer MOV AX, ptr MOV DX, ptr+2 koristi MOV EAX, ptr što je, zajedno sa kasnijim manipulacijama ovom adresom, znatno kraće i brže.

S obzirom da bilo kakve ozbiljnije primene računara danas teško mogu proći sa nekim XT-om, prekidač /G2 se može prilično bezbedno koristiti (uz obaveznu proveru na kojoj mašini program radi, inače će se na XT-u program jednostavno "zaglaviti") i tako dobiti nešto kraći i za nijansu efikasniji program. Ako računate na određeni segment tržišta, ni /G3 nije bez smisla, ali ne ove opcije ipak ne treba gledati preterano ozbiljno.

Nenad Batočanin

PREGLED DATOTEKA

Pregled *ASCII* datoteke proizvoljne dužine iz *Clipper*-a je, reklo bi se, večito aktuelan problem. Skoro da ne postoji bolja biblioteka koja na neki način ne rešava ovaj problem. Jedno od rešenja je nedavno prikazano u „Računarima" u tekstu o štampanju izveštaja iz *Clipper*-a: pregled je rešen preko asemblerske funkcije koja osim standardnog pregleda datoteke interpretira i kontrolne simbole kako bi se na ekranu videli i atributi teksta (bold, large...). Rešenje takvog problema korišćenjem isključivo *Clipper*-a nije moguće zbog brzine izvršavanja. Međutim, ako se vrši pregled čiste *ASCII* datoteke i nije potrebno vršiti analizu kontrolnih kodova, ispostavlja se da se problem sasvim lepo može rešiti i u *Clipper*-u.

Na slici 1 je prikazana procedura **List** koja omogućava pregled *ASCII* datoteke proizvoljne dužine. Procedura je maksimalno očišćena i pojednostavljena kako bi se lakše prepravljala za specifične primene. Pregled se vrši preko celog ekrana (izuzimajući poslednju liniju), ali nema nikakvih prepreka da to bude i u nekom unapred definisanom prozoru. Osnovna ideja je da se sadržaj ekrana čuva u nizu **aScreen**, a u nizu **aPos** se čuvaju adrese odgovarajućih linija u datoteci. U promenljivoj **nCol** se čuva početna pozicija prozora na ekranu. Jednostavnom izmenom ove promenljive postiže se skrolovanje levo i desno. Na listingu su implementirane komande za pomeranje za po jednu poziciju, ali je veoma prosto dodati pomeranje za više pozicija: jednostavno se pozovu odgovarajuće funkcije potreban broj puta (tasteri <**Tab**>, <**Shift-Tab**>, <**Home**>, <**End**>).

Pretpostavlja se da je datoteka koja se pregleda standardna datoteka čiji se svaki red završava parom kodova <**CR**><**LF**> i da nijedan red nije duži od unapred definisane vrednosti – pošto se ovakve datoteke uobičajeno generišu iz programa, ovo obično nije problem. Srce procedure čine funkcije za čitanje sledećeg i prethodnog reda u datoteci. Ove dve funkcije rade po sličnom algoritmu. Na primer, funkcija za čitanje sledeće linije na osnovu podataka u nizu **aPos** čita sledeću liniju iz datoteke, „skroluje" nizove **aScreen** i **aPos**, upisuje učitanu vrednost u poslednje elemente ovih nizova i na kraju se ispisuje kompletan ekran. Time se realizuju funkcije za pomeranje ekrana gore i dole, na osnovu čega se trivijalno realizuju operacije pomeranja za stranu gore/dole.

Nešto veći problem su operacije pomeranja na početak i naročito kraj datoteke. Doduše, skok na početak lako se postiže ponovnom inicijalizacijom programa. Skok na kraj je prilično nezgodan: najjednostavnije rešenje je da se uzastopno poziva funkcija **GetLine** dok pokazivač ne dođe do kraja datoteke. Ovo rešenje primenjuju mnogi programi za pregled datoteka. Međutim, takvo rešenje je veoma problematično – ako datoteka ima par megabajta, izvršavanje će potrajati minut-dva. Opet, pisanje analize datoteke „od kraja" nije preveliki problem, ali za tim nema potrebe. Primenićemo tipičnu matematičku metodu i svešćemo problem na prethodni: pretpostavimo da je nekim načinom izvršen skok na kraj datoteke i da se pritiska taster <**Down**>. Tako će prazne linije postepeno popunjavati prostor počev od poslednjeg reda i na kraju će čitav ekran biti prazan. Teorijski gledano, čitava datoteka je tada „iznad" ekrana. Ovu situaciju je lako simulirati: jednostavno treba isprazniti ceo bafer (**aScreen**) i pomeriti pokazivač na kraj datoteke.

Ako se tada (teorijski) pritisne taster <**Up**>, na vrhu ekrana će se pojaviti poslednja linija u datoteci, ponovnim pritiskom sledeća itd. Znači, dovoljno je da određen broj puta pozovemo funkciju **PrevLine** i tako smo dobili traženi kraj datoteke!

Opisano rešenje pokazuje sasvim zadovoljavajuće „brzinske" rezultate koji, što je najvažnije, ne zavise od datoteke koja se pregleda. Dodatnim optimizacijama se još može dobiti na brzini, ali i ovakvo rešenje je sasvim zadovoljavajuće. Na primer, može se povećati brzina ispisa ekrana prilikom skrolovanja gore/dole: umesto ponovnog ispisivanja čitavog ekrana može se ispisati samo nova linija, a ostatak ekrana se skroluje funkcijom **Scroll**.

PROBLEMI SA PREVODIOCEM

Koncepcija pretprocesiranja koja je primenjena u *Clipper*-u ima dosta prednosti koje smo videli mnogo puta, ali na žalost ima i određenih mana. Naime, pretprocesor obezbeđuje sloj između programera i prevodioca koji omogućava korišćenje moćnih konstrukcija, ali istovremeno (kao i svaki međusloj) otežava međusobno razumevanje dve krajnje tačke. Kad se na ovo dodaju i propusti koji su učinjeni prilikom

Listing

```
#include "inkey.ch"

#define CRLF Chr(13) + Chr(10)
#define EOF Chr(26)

STATIC aScreen, nCol, aPos, nMaxLine
STATIC nMaxCol, nHandle

*************
*
* DATUM  21.12.1993
*
* NAZIV  List (cFile)
*
* OPIS   Pregled ASCII datoteke
*
*
*****************

PROC List (cFile)
   LOCAL i, Ch, cBuffer

   f := FOpen (cFile)
   ListInit ()

   WHILE .T.

      Ch := Inkey(0)

      DO CASE

         CASE Ch == K_ESC ; EXIT

         CASE Ch == K_LEFT .AND. nCol > 1
            nCol-- ; DispScr ()

         CASE Ch == K_RIGHT .AND. nCol < nMaxCol
            nCol++ ; DispScr ()

         CASE Ch == K_DOWN
            NextLine() ; DispScr ()

         CASE Ch == K_PGDN
            FOR i := 1 TO nMaxLine
               NextLine ()
            NEXT
            DispScr ()

         CASE Ch == K_UP .AND. aPos[1] <> 0
            PrevLine ()
            DispScr ()

         CASE Ch == K_PGUP
            FOR i := 1 TO nMaxLine
               PrevLine ()
            NEXT
            DispScr ()

         // Top
         CASE Ch == K_CTRL_PGUP
            ListInit () ; DispScr ()

         // Bottom
         CASE Ch == K_CTRL_PGDN
            nPos := FSeek (f, 0, 2)
            AFill (aPos, 0) ; aPos[1] := nPos
            FOR i := 1 TO nMaxLine
               PrevLine()
            NEXT
            PrevLine() ; NextLine()
            DispScr ()

      END CASE
   END DO
   FClose (f)

RETURN

*************************
*
* DATUM  8.11.1993
*
* NAZIV  c := GetLine ()
*
* OPIS   Učitava jednu liniju
*
*
********** **** *****

STATIC FUNC GetLine ()
   LOCAL cBuffer, nRead

   cBuffer := Space(nMaxLen)
   nRead := FRead (f, @cBuffer, nMaxLen)

   IF (nPos := At(CRLF, cBuffer)) <> 0 .OR.;
      ((nPos := At(EOF, cBuffer)) <> 0 .AND. nPos == 1)

      cBuffer := PadR(Left(cBuffer, nPos - 1), nMaxLen)
      nPos := FSeek (f, (nPos + 1) - nRead, 1)

   ELSEIF nRead == 0 .OR. Left(cBuffer, 1) == EOF
      cBuffer := ""
   END IF

RETURN cBuffer

************ ********
*
* DATUM  8.11.1993
*
* NAZIV  DispScr ()
*
* OPIS:  Ispisuje ceo ekran
*
*
******** - ******

STATIC PROC DispScr
   LOCAL i := 1

   WHILE i <= Len(aScreen) .AND. NextKey() == 0
      SetPos (i - 1, 0) ; DispOut (SubStr(aScreen[i++], nCol, 80))
   END DO
   HelpLine()

RETURN

*********************
*
* DATUM  8.11.1993
*
* NAZIV  l := NextLine()
*
* OPIS:  Učitava sledeću liniju
*
*
***********************

STATIC PROC NextLine
   LOCAL lRet := .F.

   nPos := FSeek (f, aPos[nMaxLine+1], 0)
   cBuffer := GetLine ()

   IF Len(cBuffer) <> 0
      ADel (aScreen, 1)
      ADel (aPos, 1)
      aPos[nMaxLine+1] := nPos
      aScreen[nMaxLine] := cBuffer
      lRet := .T.
   END IF

RETURN lRet

*******************
*
* DATUM  8.11.1993
*
* NAZIV: PrevLine ()
*
* OPIS   Prethodna linija
*
*
*********************

STATIC PROC PrevLine
   LOCAL cBuffer, i, nBeg

   IF aPos[1] <> 0
      cBuffer := Space(nMaxLen)
      nBeg := FSeek (f, Max (0, aPos[1] - nMaxLen), 0)
      nRead := FRead (f, @cBuffer, Min (nMaxLen, aPos[1]))
      cBuffer := Left(cBuffer, nRead - 2)

      nPos := RAt (CRLF, cBuffer)

      AIns (aPos, 1)
      AIns (aScreen, 1)
      IF nPos == 0
         aScreen[1] := PadR(cBuffer, nMaxLen)
         aPos[1] := nBeg
      ELSE
         aScreen[1] := PadR(SubStr(cBuffer, nPos+2), nMaxLen)
         aPos[1] := nBeg + nPos + 1
      END IF
   END IF

RETURN

***************************
*
* DATUM  8.11.1993
*
* NAZIV  HelpLine ()
*
* OPIS   Ispisuje help liniju
*
*
****** ** ***** ****

STATIC PROC HelpLine
   @ 24, 0 SAY FILE LIST COLOR "r/w"
RETURN

*********************
*
* DATUM  3.1.1994
*
* NAZIV: ListInit()
*
* OPIS   Inicijalizacija
*
*
*********************

STATIC PROC ListInit
   ...
   nCol := 1
   nMaxCol := nMaxLen - 80
   nLine := 0
   aScreen := {}
   aPos := {0}
   FSeek (f, 0)
   WHILE nLine < 24 .AND. Len (cBuffer := GetLine ()) <> 0
      AAdd (aScreen, cBuffer)
      AAdd (aPos, nPos)
      nLine++
   END DO
   nMaxLine := Len(aScreen)
   DispScr ()
```

projektovanje prevodioca dolazi se do prilično nepri jatnih problema. Ponekad sasvim banalni uzrok može biti uzrok dugog traženja greške. Pogledajte sledeći primer

```
IF X > 0
  ? 1
ELSEIF X = „A"
  ? 2
END IF
```

Naravno, očigledna greška je što se u jednoj grani IF naredbe promenljive **X** tretira kao broj (ispituje se da li je X > 0), a u drugoj se ispituje jednakost dve stringa. Ako se za **X** zada broj koji je veći od nule, sve će lepo proći i kao rezultat će biti izvršene naredbe **? 1**. Ako se za **X** zada neke negativne vrednosti, izvršiće se druga grana **IF** naredbe i biće prijavljena greška. Međutim, greška **neće** biti prijavljena u liniji **ELSEIF** već u liniji gde počinje naredba **IF**! Ovo je donekle razumljivo (jer se cela IF naredba tretira kao jedna konstrukcija), ali u složenijim programima može biti uzrok veoma nezgodnih problema. Znači, ako dobijete grešku u prvoj liniji neke **IF** naredbe, obavezno pretražite sve uslove te naredbe. Ono što je interesantno je da ako umesto **IF** naredbe koristite **CASE** naredbu sve će biti u redu, greška će biti prijavljena u onom redu gde se i desila.

BROJ OTVORENIH DATOTEKA

Složeniji programi koji rade sa puno datoteka redovno imaju probleme sa maksimalnim brojem otvorenih datoteka. Kada se izračuna maksimalan broj otvorenih datoteka za prosečan *Clipper* program, retko se dobija brojka manja od 30 (overleji, **DBF** tabele, indeksi...). Jedini lek za ovaj problem je izmena direktive **FILES** u **CONFIG.SYS** datoteci. Logično je ba u slučaju manje vrednosti program izda upozorenje i stane. Međutim, kako odrediti iz programa da li je navedena dovoljna vrednost u direktivi **FILES**? Evo rešenje koje za to predlaže *Nantucket*

```
FUNC Handles
  LOCAL I, J, nHandle, aHandles := {}
  FOR I := 1 TO 250
    IF (nHandle := FOpen („NUL")) == -1
      EXIT
    ELSE
      AAdd (aHandles, nHandle)
    END IF
  NEXT
  FOR J := 1 TO I-1
    FClose (aHandles[J])
  NEXT
RETURN I-1
```

Ideja je veoma prosta i sastoji se u iterativnom korišćenju funkcije **FOpen** koja otvara datoteka sve dok se ne dostigne maksimalni broj. Kako bi se izbegli problemi oko otvaranja postojećih datoteka i sl. koristi se datoteka **NUL** koja uvek postoji. Funkcija prvo „otvori" koliko je moguće **NUL** datoteka, pri čemu hendle upisuje u niz **aHandles**. Kasnije se na osnovu tog niza zatvaraju sve otvorene datoteke. Rezultat rada funkcije je maksimalni raspoloživi broj hendlova.

Nastavak sa strane 111

fontovi tog dokumenta nisu čak ni instalirani. Sve 44 fonta iz prvog Microsoft-ovog *TrueType Font* paketa imaju ove osobine.

Ugrađivanje ili *Font Embedding*, omogućava da font bude uključen u sastav doumenta uz minimalnu promenu veličina datoteke. Recimo da je reč o ćiriličnoj verziji Times fonta (CTimes) koje se često sreće. Takav dokument može se otvoriti na mašini koja nema instaliran CTimes font, ako aplikacije podržava ugrađivanje fontova, moći će da prikaže i odštampa dokument sa potpuno očuvanim prvobitnim izgledom i formatiranjem. U specijalnom slučaju, činjenica da je font ugrađen u dokument pokrenuće i instalaciju ugrađenih fontova na ciljnoj mašini. Nakon toga se dokument može menjati, a CTimes koristiti i za pisanje drugih dokumenata bez narušavanja kopirajta autora i firme koja font prodaje.

Microsoft je odredio četiri nivoa ugrađivanja:
- fontovi koji se ne mogu ugraditi
- ugrađivanje koje omogućava da se font koristi samo za pregled sadržaja dokumenta (*Preview*) i štampanje, takođe poznato kao *Read-Only Embedding*
- ugrađivanje koje omogućava korišćenje ugrađenog fonta i editovanje dokumenata (*Read Write Embedding*)
- ugrađivanje uz instalaciju fontova (*installable fonts*) gde aplikacije fontove ugrađene u dokument automatski instalira u *Windows* okruženje i korisniku daje trajnu licencu za njihovo korišćenje

Nezgodno je što sve ovo vreme fontovi imaju dobar deo navedenih osobina, ali nema aplikacije koje bi te osobine podržavale. Razlog je prilično banalan: korisnici nisu znali za ove mogućnosti pa ih nisu ni zahtevali od proizvođača. Štaviše, čak se u uputstvu za *Microsoft TrueType Font Pack* tvrdilo da se fontovi mogu koristiti samo na jednom računaru, što je u suprotnosti sa idejom samog Microsofta o lakom prenošenju fontova. Nije čudo da se proizvođači softvera nisu utrkivali u korišćenju ugrađivanja fontova kada ih je sam Microsoft tako preplašio.

Prva aplikacija koja je koristila mogućnost ugrađivanja fontova bio je *Microsoft Power Point*: u **File/Save As** dijalogu pojavila se opcija **Save With Fonts**. Kada se uključi ova opcija, svi korišćeni fontovi se snimaju zajedno sa dokumentom i mogu se koristiti na bilo kojem računaru na kojem se izvršava *Power Point*. Međutim, desila se komunistička diverzija u Microsoftu, neko tamo je ljubitelj principa „jedan korak napred, dve korake nazad" pa se u *Power Pointu 3.0*, najnovijoj verziji paketa, mogu koristiti samo fontovi koji omogućavaju instalaciju na ciljnom računaru. Korišćenje *read-only* i *read-write* fontova sačekaće sledeću verziju.

Druga aplikacija je *Word for Windows 6.0*: isprobao sam ugrađivanje fontova i sve lepo funkcioniše. Upozorenje za ovaj slučaj: nemojte poverovati Microsoftu da ugrađivanje neznatno povećava dužinu dokumenta!

Još malo o Windowsu 3.11

Naravno, *Windows for Workgroups 3.11* je odavno među nama, zahvaljujući singapurskim i inim kanalima za nabavku softvera. Interesantno je da se paket može komotno koristiti i bez mrežnih opcija: jednostavno pri prvoj instalaciji ili ažuriranju stare izaberete da se ne instalira podrška za rad na mreži. Time ćete i na individualnoj kućnoj mašini dobiti unapređenja iz nove verzije, kao što je 32-pristup datotekama. Ostali interesantni noviteti pogodni i za kućnu upotrebu su *Remote Access*, za vezu sa mrežom na drugoj lokaciji, i mogućnost slanja i prijema faks poruka. Ovo drugo je primena nove tehnologije *Microsoft At Work*.

Kao i do sade, WIN.COM se može pokrenuti sa nekim opcijama. Evo novih opcija, koje se odnose na 32-bitni pristup disku i datotekama, odnosno na mrežne sposobnosti:

/N	pokretanje Windowsa bez mrežnih drajvera
/D	C isključuje 32 bitni pristup datotekama
	F isključuje 32 bitni pristup disku – ekvivalentno unošenju opcije 32BitDiskAccess=FALSE u SYSTEM.INI datoteci

Pavle Peković

Unix Shell

Kada otvaram račun novom korisniku na nekoj od Unix mašina koje administriram, prvo ga pitam za željeni *username*. U većini slučajeva ljudi znaju o čemu se radi, a i onima koji ne znaju vrlo lako objasnim. Na pitanje o imenu i prezimenu, naravno, odmah daju odgovor. Izbor *password*-a, kao i kucanje istog dva puta radi provere, potraje izvesno vreme ali ipak i to prođe manje-više bezbolno. Ali kada ih upitam koji *shell* žele, tu me većina zbunjeno pogleda a u očima im čitam poruku "System halted". Zato već neko vreme preskačem to pitanje i stavljam *bash* za osnovni *shell*, pa ako im se ne dopada neka mi pošalju *mail* nije mi teško da promenim. Do sada sam *mail* sa zahtevom za promenu *shell*-a dobio svega dva ili tri puta. Da li je uzrok tome što je *bash* toliko dobar ili što korisnici ne znaju za bolje, nisam proveravao. Rizikujući da ću ubuduće dobijati više *mail*-a sa zahtevom za promenu *shell*-a i trošiti više svog vremena, pokušaću da pojam *shell* spustim među narod.

Kada radite za nekom Unix mašinom, ona od vas očekuje da zadate komandu i definišete eventualne parametre na osnovu kojih sistem izvršava odgovarajuću akciju. Ovu interakciju sa korisnikom obavlja posebna vrsta programa koja se uobičajeno naziva *shell* ili kako se u nas uobičajeno prevodi ljuska, školjka. Iako je osnovna ideja veoma jednostavna, ja ni posle sat vremena smišljanja kako da na ovoj jednoj strani napišem samo ono najbitnije o *shell*-ovima to nisam uspeo da uradim. Zato bi se ovaj nastavak "Moje škole *Unix*-a" mogao nazvati "Moja škola *Unix shell*-ova".

Kada se prijavite na *Unix* mašinu, nakon uobičajenih pozdravnih poruka dobićete *shell* prompt (znak $, %, # ili sl.), što označava da je *shell* spreman da interpretira vaše komande. Počnimo od onih najjednostavnijih. Otkucajte **who**. *Shell* će startovati program *who* i njegove rezultate (lista korisnika trenutno prijavljenih na sistemu) će se ispisati na izlaznom uređaju. Nakon toga otkucajte nazive još nekoliko programa za koje znate da postoje, a *shell* će vredno analizirati zadate komande i interpretirati ih na odgovarajući način (startovaće zadate programe).

Sada otkucajte recimo **echo Zdravo**. Na ekranu će se pojaviti reč "Zdravo". Po prethodnoj logici *shell* je izvršio program *echo* sa parametrom *Zdravo*. Međutim, to nije tačno. Program *echo* ne postoji, *shell* je izvršio svoju internu komandu koja prikazuje zadati tekst na ekranu.

Dakle, *shell* je program koji na zahtev korisnika izvršava svoje interne komande i eksterne programe. A šta ako nam je potrebna nekakva veza između izvršavanja komandi koje zadajemo? Recimo, da se izvrši program B samo ako se uspešno izvrši program A ili pak da se izvršavanje programa B ponavlja sve dok izlazni rezultati ne budu onakvi kakve tražimo. I za to postoji rešenje. Svi poznatiji *shell*-ovi poseduju kompletan set komandi za podršku strukture selekcije (**if, case**...) kao i za iterativne strukture (**do, while, for**...). Da bi sistem bio kompletan, tu su i tzv. **shell** i **environment** promenljive (posmatrajte ih kao promenljive u bilo kom programskom jeziku) kao i mnogi drugi elementi sa kojima ćemo se upoznati kasnije.

Pored ovog, postoje još brojni značajni elementi *shell*-a. Jedan od njih je svakako i editor komandne linije. Ovaj editor omogućuje korisniku da zada komandu uz što veću fleksibilnost pri unosu. Kvalitet editora komandne linije se meri po tome da li sadrži sledeće elemente:

- ispravljanje pogrešno otkucanih delova komande
- ponavljanje prethodno unetih komandi (*history*)
- definisanje kratkih imena za često korišćene komande (*alias*)
- dovršavanje jednoznačno definisanih imena datoteka, promenljivih i drugih skupova imena

Od značajnih osobina shell-a pomenimo još i redirekciju, tj. mogućnost da se izlazni rezultati jednog programa koriste kao ulazni drugog, zapisivanje rezultata koji se prikazuju na ekranu u datoteku, uzimanje vrednosti iz datoteke kao ulaznih parametara za neki program i sl.

Shell-ova ima mnogo, ali najpoznatiji su **sh**, **csh** (*shell* sa *C*-like sintaksom), **bash** (unapređeni **sh**), **tcsh** (unapređeni **csh**), **ksh** (**Korn shell**) itd. Njihove mogućnosti, kao i mane i vrline upoznaćemo kroz nekoliko narednih nastavaka "Moje škole Unix-a". Tada ćete, nadam se, moći da ocenite koji *shell* najviše odgovara vašim potrebama.

Tabela 1

sh, bash	csh, tcsh	
PATH	path	Definiše listu direktorijuma koja predstavlja put po kome će shell tražiti zadati program
HOME	home	Definiše osnovni direktorijum korisnika. Ovo je bitno za komandu cd koja, kada se zada bez argumenata, koristi vrednost promenljive HOME tj. home i prelazi u direktorijum čiji naziv sadrži ova promenljiva
MAIL	mail	Definiše put i naziv datoteke koja sadrži nove email poruke. Ako je ova varijabla definisana, shell nakon završetka svake zadate komande proverava tu datoteku da li sadrži nove email poruke i ispisuje odgovarajuću poruku (npr. You have new mail)
CDPATH	cdpath	Slično kao PATH tj. path, s tim što se lista direktorijuma koristi kao put za komandu cd
PS1	prompt	Definiše izgled prompta. Ako recimo koristite bash i ako želite u promptu tekući path, kucajte PS1='$PWD > ' export PS1

Shell i environment promenljive

Environment promenljive su set promenljivih koji "vidi" svaki pokrenuti program, pa čak i *shell*. Sa druge strane, *shell* promenljive su promenljive koje su vezane za određeni *shell*, s tim što vrednost za deo tih promenljivih konkretan *shell* koristi iz vrednosti *environment* promenljivih. Tako na primer *shell* promenljiva path iz csh pokazuje na vrednost environment promenljive PATH i svaka promena shell promenljive path menja i vrednost environment promenljive PATH. U ovom slučaju važi i obratno, mada to nije pravilo (na primer varijable prompt (**csh**) i PS1). Da bi razlika između *shell* i *environment* promenljivih bila jasnija, pretpostavimo da je vaš osnovni *shell* (*login shell*) **csh**. Definišite *shell* promenljivu **shell_prom** komandom

set shell_prom=neki_string

a zatim definišite *environment* promenljivu ENV_PROM komandom

setenv ENV_PROM neki_string

Pogledajte listu definisanih *shell* promenljivih komandom **set**, kao i listu *environment* promenljivih komandom **env**. Nakon ovog startujte novi program i to ne bilo koji, nego baš neki od drugih *shell*-ova, na primer **sh**. Bilo da otkucate **set** ili **env**, videćete da u listi promenljivih nema **shell_prom** promenljive jer je ona definisana samo za prethodni *shell* (**csh**), dok je *environment* promenljiva ENV_PROM i dalje tu sa prethodno definisanom vrednošću. *Environment* promenljive koristi veliki broj programa i one im služe za interno konfigurisanje. Tako na primer, ako vam se u *elm mailer*-u ne dopada *default* editor, postavite *environment* promenljivu EDITOR u koju ćete upisati put i ime editora koji želite da koristite. Ako je to editor *cr* koji se nalazi u **/usr/local/bin** direktorijumu, za *sh* ili *bash* kucajte

EDITOR=/usr/local/bin/cr
export EDITOR

dok za *csh*, *tcsh* treba otkucati

setenv EDITOR /usr/local/bin/cr

Shell varijable se koriste uglavnom za konfigurisanje samog *shell*-a. Tako postoji veliki broj unapred definisanih varijabli, kao i brojne rezervisane, čijim se promenama vrednosti menja ponašanje *shell*-a. Verovatno je prva promenljiva koju korisnik izmeni PATH (**sh, bash**) ili **path** (**csh, tcsh**) varijabla. One se koriste za određivanje puta kroz direktorijume kojim će *shell* tražiti zadatu komandu. Ako vam je teško da među brojnim varijablama prikazanim komandama **set** ili **env** tražite onu koja vam je potrebna, njenu vrednost možete dobiti tako što ćete otkucati

echo $naziv_promenljive

Dakle

echo $PATH

daje trenutnu vrednost promenljive PATH. Vrednost promenljive se menja na već opisan način, dok recimo, ako želite da dodate neki direktorijum (na primer **/usr/local/bin**) u listu direktorijuma u promenljivoj PATH, za **csh** ili **tcsh** kucajte

set path=($path /usr/local/bin)

a za **sh** ili **bash**

PATH=$PATH:/usr/local/bin

Pošto je znak *space* separator komandi i argumenata u komandnoj liniji, a vrlo često je potrebno da upravo *space* bude jedan od znakova u vrednosti promenljive, u tim slučajevima je vrednost potrebno staviti pod znake navoda, apostrofe ili samo u **csh** i **tcsh** zagrade. Tako će, na primer, u **sh** komanda

IME=Branka Radulovic

proizvesti grešku. Korektna dodela vrednosti se obavlja na sledeći način

IME="Branka Radulovic"

ili

IME='Branka Radulovic'

Svejedno je da li koristite apostrofe ili navodnike u ovim primerima. Međutim, pogledajmo sledeća primera

PORUKA="Nove poruke su u $MAIL"

i

PORUKA='Nove poruke su u $MAIL'

Nakon prve dodele string $MAIL će biti zamenjen vrednošću promenljive MAIL, dok to u drugom primeru neće biti slučaj.

Sigurno ste primetili da se *environment* promenljive uglavnom pišu velikim slovima, dok to pravilo za *shell* varijable zavisi od samog *shell*-a. Tako se u **csh** i **tcsh** *shell* promenljive uobičajeno pišu malim slovima. Takođe, kada definišete promenljive u **sh** ili **bash**, da bi se ta promenljiva "videla" i u svim potprocesima tekućeg *shell*-a (deca procesi), dodajte obavezno

export ime_promenljive

Pored pomenute PATH tj. *path* (**csh, tcsh**) *shell* promenljive, korisniku su na raspolaganju još brojne druge promenljive uz pomoć kojih može *shell* prilagoditi svojim potrebama. Pregled značajnijih *shell* promenljivih je dat u tabeli 1.

Naravno, pored ovih "opštih", svaki *shell* poseduje i neke specifične varijable koje pružaju veće mogućnosti u konfigurisanju *shell*-a. U tome prednjače **bash** a naročito **tcsh**. Sve u svemu, svi *shell*-ovi, uz izvesne razlike, na podjednako jednostavan način manipulišu *shell* i *environment* promenljivama. Ono u čemu **tcsh** prednjači je veliki izbor varijabli za konfigurisanje. Dakle, do sledećeg broja Računara poen za **tcsh**, pola poena za **bash**.

ppekovic@sezam
paya@fon.fon.bg.ac.yu
epekovic@yubgef51.bitnet

817 [OS.MSDOS, PC] DOS funkcija FIND FIRST (4Eh) vraća status da fajl postoji i ako se navede neki od postojećih karaktera device drajvere – FIND FIRST(„con") na primer vraća da je sve u redu i da postoji fajl se tim imenom. Atribut tog fajla je postavljen na 40h. Zanimljivo je da je to vrednost atributa nedokumentovana, ne samo u raznim DOS priručnicima nego i na najnovijem *Microsoft*-ovom CD ROM-u sa tehničkim podacima o DOS-u i srodnim operativnim sistemima.

„Oficijelni" način da se proveri da li neko ime odgovara drajveru je da se fajl otvori, pa zatim testira odgovarajućom funkcijom. Taj metod nije pogodan u mrežnom okruženju – FIND FIRST je mnogo jednostavnije rešenje, ako bi se moglo verovati da će vrednost 40h opstati i u budućim verzijama DOS-a [prilog **Zoran Živetić**]

818 [OS.MSDOS, PC] Već smo govorili o izmenama u funkcijama za formatiranje koje je *Microsoft* uneo u DOS 6.2. Jedna od zanimljivih novosti pokazuje se u tretmanu loših sektora. Uzmite formatiranu i ispravnu disketu, bez loših sektora, startujte *Norton Utilities* ili neki sličan program i obeležite neki sektor na njoj kao loš, iako je u stvari dobar. Onda izvršite FORMAT A: /U i sektor će ostati loš. To praktično znači da /U ne označava baš sasvim „bezuslovno" formatiranje. Za ponovno testiranje loših sektora koristi se opcija /C, na primer FORMAT A: /C /U.

Ova izmena je u suštini dobra – ako je neki sektor u nekom trenutku bio sumnjiv, možda je bolje da ostane obeležen kao loš iako nije loš, nego da bude obeležen kao dobar pa da posle upropasti neke podatke. Jedino je nevolja što malo ko zna za opciju /C.

819 [OS.MSDOS, PC] Kada smo već kod DOS-a 6: Da li ste primetili da podrazumevani prompt više nije $g nego pg, dakle čak i ako ne postoji AUTOEXEC.BAT, dobićete u promptu ime tekućeg direktorijuma: C:\> umesto ružnog C>. Zanimljivo je, sa druge strane, da će po prelasku na disk A prompt biti A> a ne A:\>, što znači da komanda PROMPT pg i dalje nije izgubila smisao [preuzeto sa Sezama, prilozi **Dragan Grbić** (*dr grba*) i **Čedomil Višnjić** (*fancy*)]

820 [OS.WINDOW, PC] Kako da čitav *Windows 3.1* ili bar ono što je potrebno da bi on normalno radio smestite na jednu AT (1.2 megabajta) disketu i da na njoj ostane praznog mesta? Zvuči nemoguće ali nije, samo treba tačno znati šta je koja datoteka i može li se bez nje. Na slici 1 dat je spisak fajlova koje treba smestiti na disketu [preuzeto sa Sezama, prilog **Zoran Milosavljević** (*zoran*)]

821 [OS.WINDOW, PC] Kompjuter se sve češće koristi kao namenska alatka koja radi samo jedan posao – neko ko, na primer, stalno radi u *Corel*-u može smatrati neprijatnom obavezom startovanje *Windows*-a, aktiviranje odgovarajućeg foldera, dvostruki klik na ikonu... Sve se može automatizovati – u [boot] sekciju datoteke SYSTEM.INI, umesto SHELL=PROGMAN.EXE treba upisati SHELL=CORELDRW.EXE. Kada se radi o napuštanju *Windows*-a, vredi preuzeti sa Sezama program WINEXIT, pa mu u komandnoj liniji (u **Properties**) dodeli jedno QUICK. Kasnije se izlazi klikom na odgovarajuću ikonu, bez Alt F4 i ostale „gnjavaže" [preuzeto sa Sezama, prilozi **Nenad Milanković** (*wizard*) i **Novica Milić** (*novim*)]

822 [OS.MSDOS, PC] Više puta je bilo reči o .BAT proceduri koja bi proverila da li neki direktorijum postoji ili ne, naravno bez primene kakvih alata osim samog MS DOS-a. Ranije predloženo rešenje, korišćenje komandi TREE i FIND, uradi posao ali je vrlo neelegantno. No, sledeća procedura savršeno rešava problem – zasniva se na datoteci NUL koja postoji u svakom direktorijumu i komandi EXIST koja neće proizvesti grešku u problematičnoj situaciji:

```
@echo off
if „%1"=="\" goto ima
if exist %1\nul goto ima
echo %1 ne postoji
goto kraj
:ima
echo %1 postoji
:kraj
```

[preuzeto sa Sezama, prilog **Milovan Kovačević** (*majki*)]

823 [OS.MSDOS, PC] Kako iz komandne linije obrisati datoteku koja ima neke specifične atribute (*read only, hidden*...)? Možete da se mučite sa komandom ATTRIB, ali je mnogo lakše da otkucate DELTREE ime.tip. Nije baš dokumentovano, ali radi! [preuzeto sa Sezama, prilog **Dejan Vesić** (*dvesic*)]

824 [HARDW, PC] Mala čarolija koja ponekad može da pomogne vlasnicima prenosnih računara – ako vaš sistem iznenada počne da zastaje pri testiranju memorije, isključite računar, odvojite tastaturu, okrenite je naopako i dobro protresite, ne bi li svo „đubre" iz nje ispalo. Ponovo povežite tastaturu i... memorija je OK. Tajna je u tome što mnogi portabl računari koriste deo *keyboard encoder* čipa za upravljanje adresnom linijom 20, preko koje se komunicira sa visokom memorijom. Ukoliko negde u tastaturi postoji kratak spoj, računar će imati problema i sa pristupom memoriji, a pošto se memorija testira pre tastature, prijaviće grešku i prekinuti rad! [preuzeto sa BIX-a]

825 [SIS, PC] Odakle da pročitate datum kreiranja BIOS-a koji se, na primer, može ugraditi u neku zaštitu? String je na F000:FFF5-FFFC [preuzeto sa Sezama, prilog **Miloš Đermanović** (*eagle*)]

826 [PJ.PASCAL, PC] Velika mana *Turbo Pascal*-a je što ne može da pravi .OBJ fajlove koji mogu da se povezuju sa drugim aplikacijama. No, ponekad se možete poslužiti i malim trikom: modifikujte glavni .PAS program i ubacite na početak INTERFACE i IMPLEMENTATION sekciju, kao da se radi o junitu (*unit*). Prevedite ga pa, koristeći program TPU2ASM koji možete da preuzmete sa Sezama, konvertujte .TPU u .ASM. Zatim, koristeći bilo koji asembler, prevedite .ASM u .OBJ. Na žalost, program sa Sezama radi samo za junite koji su kompajlirani verzijom 5 kompajlera – za novije bi trebalo registrovati paket kod firme *TurboPower* [preuzeto sa Sezama, prilog **Nenad Nikolić** (*nikolaus*)]

827 [PJ.PASCAL, PC] Kako se u memoriju VGA karte učitavaju definicije karaktera, recimo YU slova? Pogledajte program sa slike 2 – on uzima podrazumevani font 8×16, zamenjuje slova A i B i aktivira novi font [preuzeto sa Sezama, prilog **Miloš Đermanović** (*eagle*)]

828 [TEHP, PC] Na slici 3 dat je kratak i pouzdan program koji proverava da li je miš prisutan ili ne. Iako je pisan za Borlandove kompajlere, uz male prepravke može se primeniti i na bilo kom drugom C-u [preuzeto sa BIX-a]

829 [TEHP.PASCAL, PC] Pri radu sa *Borland Pascal*-om 7.0 u zaštićenom režimu pojavljuje se problem sa nekim DOS-ovim IOCTL funkcijama: brisanje fajla na primer radi savršeno, a čitanje ili setovanje serijskog broja diska ne! Na slici 4 dato je rešenje, preuzeto sa Borlandovog BBS-a, kako se iz DPMI programa poziva neki interapt realnog moda [preuzeto sa Sezama, prilog **Predrag Dobrević** (*predragd*)]

830 [BASE.CLIPPER, PC] U složenijim *Clipper* aplikacijama korisniku često treba omogućiti pregled nekog tekstualnog fajla, na primer izveštaja koji će se kasnije štampati ili nekog niza ulaznih podataka. Neki za to koriste poziv eksternog programa (obično po znalog *shareware* programa LIST), ali je zgodnije koristiti elemente samog *Clipper*-a: dobar kandidat je MEMOEDIT, ali se tu javlja problem ako tekstualna datoteka ima više od 64 kilobajta. Poslužite se programom sa slike 5 – koristeći *TBrowse* i funkcije za rad sa datotekama iz *NanFor* biblioteke, realizovana je kratka i veoma efikasna LIST naredba koja može da radi sa proizvoljno velikim datotekama. Poziva se se LIST (ime datoteke), na primer LIST (READ.ME). Ako se pokaže bag *NanFor* biblioteke u vidu nekog LF karaktera viška, **bCol** := TBColumnNew („", {|| FT_FReadLn() }) zamenite sa **bCol** := TBColumnNew („", {|| StrTran (FT_FReadLn(), Chr(10), „") }) [preuzeto sa Sezama, prilog **Jovan Rulejić** (*bulaja*)]

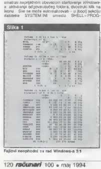
831 [BASE.CLIPPER, PC] Da li vam se, pri prevođenju ne posebno složenih *Clipper* programa, javila poruka „string space exhausted"? Problem se obično reši tako što se procedura podeli na dve ili tri, ali bolje ga je rešiti „jednom zauvek". U pitanju je, zapravo, bag *Clipper*-a 5.2 (CA), rešava se primenom *patch*-a ili prelaskom na 5.2c. *Patch* može da se preuzme sa Sezama [prilog **Zoran Đuričić** (*zorand*)]

832 [ZAST, PC] U našim krajevima se pojavio još jedan virus, koga najnoviji SCAN detektuje pod imenom *Stamford*. Virus se prenosi na računar ako ga resetujete uz zaraženu disketu u drajvu A, kao Mikelanđelo. Virus se upisuje u particionu tabelu i *boot* sektor, a docnije je rezidentan u memoriji. Nije primećeno kakvu štetu pravi, ali izgleda da se ne vezuje na datoteke. SCAN ga pronalazi, ali se viđene verzije ne mogu odstraniti sa CLEAN – posle CLEAN C: [Stam] dobija se poruka da se virus ne može bezbedno odstraniti. MSAV iz DOS-a 6.0 ga uopšte ne pronalazi.

Ako se napravi *backup* svega, ukinu particije, ponovo naprave i formatira disk, virus je i dalje u particionoj tabeli! Ali, ako se iz *Norton*-a ili nekog drugog sličnog programa particiona tabela napuni nulama, pa se onda kreiraju particije, virus je mrtav. Eventualno se može ukloniti i nedokumentovanom opcijom /MBR komande FDISK. Ako ste primenom odgovarajućih alatki iz paketa *Norton Utilities* ili *PC Tools* sačuvali „sliku" particiona tabele i *boot* sektora, možete ih vratiti na disk i tako uništiti virus. Pre bilo koje od ovih „terapija" preporučuje se detaljan *backup* svih relavantnih podataka.

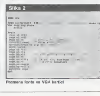
833 [OTXT.WORD, PC] Mnogi su poželeli da unaprede svoj *WinWord 6.0* u 6.0a, koristeći javno dostupni *upgrade* koji se, pod imenom WORD60A.A01-A05, može preuzeti iz Sezamovog direktorijuma. Na žalost, pokušaj pokretanja tog programa uglavnom proizvodi poruku o grešci, jer je kod nas rasprostranjena neke od beta verzije *WinWord*-a. Srećom, leka ima, u datotekama WIN.INI i SETUP.STF (a i na dru-

Slika 1



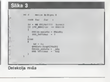
Fajlovi neophodni za rad Windows-a 3.1

Slika 2



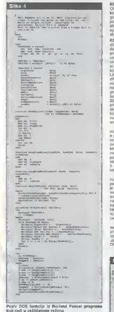
Promena fonta na VGA kartici

Slika 3



Detekcija miša

Slika 4



Poziv DOS funkcije iz Borland Pascal programa koji radi u zaštićenom režimu

gim mestima ako se spominje) treba pronaći string 2222 (interna oznaka revizije) i zameniti ga sa 2226. Kasnije startovan *upgrade* obaviće posao. Naravno, s obzirom da ga primenjujete na nešto za šta nije pisan, odgovornost za sve kasnije probleme je na vama [preuzeto sa Sezama, prilozi: **Zoran Paunović** (*peacock*) i **Jovan Bulajić** (*bulaja*)]

834 [OTXT WORD, PC] Dešava li vam se da *WinWord 6* „iz čista mira" pretvori neko malo 'i' u veliko? Pojava će vas naprosto isterati iz kože jer se u našim tekstovima slovo i vrlo često javlja kao veznik, a kada se pretvori u veliko slovo čitav tekst postane neupotrebljiv. Na sreću, može joj se „doći glave" – aktivirajte *auto correct* opciju **Tools** menija i videćete FIND i REPLACE I. Uklonite ovaj red i vaše malo i je bezbedno. „Štos" je očito u tome što se u engleskom jeziku I (koje znači „ja") piše kao veliko slovo [preuzeto sa Sezama, prilog: **Nebojša Jevtić** (*jevta*)]

835 [OTXT QUARKXPRESS, PC] Instalirali ste *QuarkXpress*, startovali ga i dobili poruku da vaša konfiguracija nije kompatibilna sa tom verzijom programa. Da kupujete novi računar? Nije nužno – samo pokrenite *Setup Windows*-a za Istočnu Evropu i u stavci „jezik" odaberite engleski. Naravno, trajno rešenje je kupovina evropske verzije *QuarkXpress*-a, ali ako je već stigao američki...

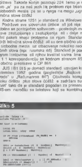
836 [OS WINDOWS, YUZI.530, PC] Sve više *Windows* aplikacija podržava ćirilicu, ali po kom rasporedu? Trenutno je u „opticaju" više varijanti. Najveći broj programa podržava kodnu stranicu 866, koja je bazirana na ruskoj ćirilici, ali podržava i beloruska i bugarska slova. U samoj Rusiji ova kodna strana je apsolutni standard na PC računarima. Postoji podrška za nekoliko novijih modela štampača (*Proprinter*, Epsonovi LX/LQ 100, LQ 570+...) kao i za *MS Word*, *WordPerfect*... Na nivou DOS-a standard je podržan komercijalnim paketom *Alphabet Plus*, kao i kod nas neobično popularnim *ParaWin*-om ruske firme *Paragraph*. Dobra strana standarda je što ostavlja 7 slobodnih mesta za dodatne znake, a loša što koristi kod 224 koji mnogim editorima pravi probleme. Može se prilagoditi srpskoj ćirilici jedino izbacivanjem nekih ruskih slova – s obzirom da naša ćirilica ima 6 slova koje Rusi nemaju, potrebno bi bilo 12 kodova (velika i mala slova), a raspolaže se samo sa 7, dakle moraju se zameniti i ruska slova.

Kodna stranica 855 pokriva sve slovenske ćirilice, izuzev jednog slova rusinske. Nije u širokoj upotrebi, a podrška je dosta slabija nego kod CP866. Pojavljuje se u nekim novijim Epson štampačima i *Microsoft*ovim paketima. Loša strana standarda je što uništava brojne znakove za okvire, tako da je pod DOS-om neupotrebljiv, iako ga pomenuti paket *Alphabet Plus* podržava. Takođe koristi poziciju 224, tamo je rusko slovo „ja", pa za naš jezik taj problem nije bitan. Ima 6 slobodnih mesta, pa se u njega ne mogu „ugraditi" latinična slova čćđšž.

Kodna strana 1251 je standard za *Windows* ćirilicu. Podržava sve slovenske ćirilice, ali još nije kompatibilna sa postojećim softverom... poslednjih meseci je sve zastupljenija i zastupljenija, ali i dalje mnogi važni paketi imaju problema sa njom. Standard ne sadrži latinična slova čćđšž, ali se ona obično ubacuju na slobodna mesta uz izbacivanje nekoliko ređe potrebnih slova (npr. rusinska ïđ). Standard je podržan brojnim fontovima, ali njihov kvalitet varira. Ima (skoro) 1:1 korespondenciju sa kodnom stranom 855, a se obično preslikava u CP 866.

JUS I B1 015 je domaći standard, usvojen krajem septembra 1992. godine (pogledajte „Bajtove lične prirode" u „Računarima 86"). Obuhvata kompletnu srpsku ćirilicu plus slova čćđšž, a zadržani su i svi „okviri", tako da je standard pogodan za primenu pod DOS-om, naročito sa tekstova koji se kombinuju od

Slika 5



LIST na Clipper-u

ćiriličnih i latiničnih segmenata. Standard je podržan DOS paketom ALL4YU 1.0 (u javnom vlasništvu), a poznat je i način da se ROM-ovi pojedinih Epson štampača prilagode ovakvom ispisivanju znakova. Postoji nešto domaćeg bibliografskog softvera koji podržava dati paket, ali je u celini jako malo onih koji ga koriste, možda i zato što nije prilagođen *Windows* okruženju [preuzeto sa Sezama, prilog: **Nenad Bogojević** (*nkbog*)]

837 [MREZ.NOVELL, PC] Šta je nova doneo *Novell NetWare 4.0*? O osnovnom poboljšanju, *NetWare Directory Service*, bilo je već reči – promene u upravljanju memorijom, kompresija i migracija podataka... No, da pomenemo i neka praktična iskustva.

Interna kompresija datoteka ne podseća na popularne DOS alatke kao što su DBLSPACE ili *Stacker*. Kompresija se, naime, ne vrši nad čitavim diskom već samo nad označenim datotekama, direktorijumima ili particijama. Kompresija se aktivira tek kada objektu (na primer datoteci) niko ne pristupa u neko (unapred zadato) vreme. Čim neko zatraži deo komprimovanog objekta, on će se automatski dekomprimovati i biti raspoloživ u punom obliku dok je potreban.

Alokacija ispod nivoa bloka (*Block Suballocation*) rešava probleme velikog broja malih datoteka na disku – „rep" fajla koji je pretekao na dodeli klaster može taj prostor podeliti sa „repom" nekog drugog fajla, tako da je *slack* minimalan. Za razliku od *NetWare 3.11* kod koga je preporučena veličina bloka 4 kilobajta, ovde se koriste „podblokovi" od 512 bajta. Veličina osnovnog bloka i dalje se zadaje pri kreiranju volumena i može da bude 4, 8 ili više (do 64) kilobajta.

Migracija podrazumeva premeštanje retko korišćenih podataka na sporije medije, tipično magnetnu traku ili CD-ROM. *NetWare 4.0* i dalje „vidi" preseljene podatke kao deo *NetWare* particije – kada korisnik zatraži podatke koji su migrirali, operativni sistem će ih automatski vratiti na hard disk servera.

Promene u upravljanju memorijom bile su neophodne počev verzije 3.11: po priznanju samog *Novell*-a, povremeno nakon završetka procesa ne oslobodi u celini memoriju koju je proces koristio, njen deo ostaje vezan za jedan *memory pool*, pa se kasnije može alocirati samo od strane „sličnog" procesa. Pokušajte, na primer, na disku od oko 200Mb gusto popunjenom malim datotekama i uz RAM od 4 M da učitate NETSHLD, PSERVER i MONITOR – verovatno ćete doživeti totalnu havariju i dobiti ponudu da se memorija snimi na disketu pre pada sistema. *NetWare 4.0* zahvata memoriju iz jedinstvenog *pool*-a koji pretenduje da reši sve probleme ovoga tipa.

Konačno je od korisnika sklonjena „šuma" redova za štampu i print servera – on sada šalje datoteku pravo na odredišni štampač. Zaobiđen je i problem štampe na serveru na koji niste prijavljeni, a administratorima je omogućena još preciznija (i zapetljanija) manipulacija redovima za štampu, print serverima, mrežnim štampačima i njihovom vezom sa korisnicima. To se obavlja pomoću uslužnog programa koji radi pod *Windows*-om.

Administratori *Novell* mreža dobili su nove mogućnosti kontrole korisnika. Ovlašćeno lice (tzv. *auditor*) može da zahteva da mu sistem dojavi sve dolaske i odlaske sa mreže, promenu prava pristupa resursima, kreiranje datoteka, direktorijuma, upis, čitanje, zahteve za manipulaciju redovima za štampu, događaje direktno povezane sa korisnicima i tome slično. Administrator mreže može ovlastiti neke manje privilegovane korisnike da budu auditori [preuzeto sa Sezama, prilog: **Vladimir Popović** (*vpopovic*)]

838 [COMM, VAX] Pri radu sa VAX-ovim konferencijskim sistemom NOTES pogodno je pri odgovaranju na poruku citirati deo pitanja. Na žalost, način da se deo teksta iz gornjeg prozora (u kome je pitanje) prenese u donji (rezervisan za odgovor) nije baš očigledan. Treba najpre pritisnuti Ctrl B, otkucati „other" i tako preći u drugi prozor. Zatim na numeričkoj tastaturi pritisnite broj 7, obeležite blok strelicama, pritisnite 8, vratite se u donji prozor (Ctrl B Enter) i umetnite tekst pritiskom na broj 9 sa numeričke tastature (pretpostavlja se da lampica Num Lock gori – ako je isključena, umesto 7, 8 i 9 treba pritiskati Shift 7, Shift 8 odnosno Shift 9). Posle toga ćete ručno na početak svakog reda dodati po znak kojim obeležavate citat i nastaviti sa kucanjem odgovora [preuzeto sa DECnet-a, autor: **Ivan Glišin** (*ubbg::eglisin*)]

Priloge za ovu rubriku šaljite na adresu „Računari" (za „Bajtove lične prirode"), Bulevar vojvode Mišića 17, Beograd, ili preko Sezama u vidu lične poruke uredniku (*mail write dejanr*)

Sezam je u martu opslužio 33,482 poziva, komunicirao sa korisnicima 7,320 časova tj preko 305 dana (u proseku je 9 8 nodova bilo neprekidno zauzeto) od čega je u *chat*-u provedeno gotovo 1320 sati ili 165 radnih dana Korisnici su sa Sezama *download*-ovali gotovo 1 7 gigabajta podataka, dok su za isto vreme sistemu poslali skoro 55 megabajta

SEZAM

SISTEM ZA MODEMSKE KOMUNIKACIJE

29

Priprema: Dejan Ristanović u saradnji sa moderatorima konferencija

Moderatori: Dejan Bajnca (CIVILIZACIJA) Nikola Bošković (PC USER) Jovan Bulajić (FILES) Dragan Čitaković (IGRE) Dejan Jelović (PC PROG) Darko Jevtović (PC HARD) Slobodan Kaležić (ORKA) Ranka Jovanović (RAČUNARI) Bojan Pavković (PC UTIL) Pavle Peković (PC OS) Dejan Ristanović (NOVOSTI i KOMUNIKACIJE) Dorian Staničić (VICEVI) Bojan Stanojević (ATARI) Bojan Tepavčević (FORUM) Zoran Životić (SEZAM)

Najviše poziva uputili su Čedomil Vranjić (*fency*) Dejan Jelović (*djelovic*) i Marko Milivojević (*markom*) najviše vremena na vezi su proveli Dragan Čitaković (cdragan) Vesna Denčić (*vesna*) i Neda Rosandić (*lelma*) na 20NAJ listi *chat* a pridružio im se i Bojan Stanojević (*bojs*) dok su datoteke najviše prenosili Dragan Krstić (*dkrstic*) Nenad Aničić (*anenad*) i Nebojša Majstorović (*nmajsor*) Najviše konferencijskih poruka poslali su Marko Milivojević (*markom*), Mihailo Stefanović (*miks*) i Srđan Pantić (*spantic*)

Kada već pomenusmo konferencije u njih je tokom marta stiglo 11312 poruka ili gotovo 12 megabajta teksta na računajući ih 5 5 megabajta datoteka uz poruke Tome treba dodati preko 11 000 privatnih poruka (8 6 megabajta teksta) i 7 167 poruka u grupama (2 8 megabajta ne računajući priključene datoteke)

SEZAM — Zoran Životić

U konferenciju SEZAM stiglo je 520 poruka znatno manje nego prethodnih meseci Nije bilo posebno zanimljivih diskusija – većina poruka bavila se tekućim događajima na Sezamu i predlozima za novu verziju softvera Sredinom marta je posle par meseci zakašnjenja iz štampe izašlo treće izdanje Uputstva za upotrebu Sezama što su korisnici naročito oni koji su se nedavno prijavili i koji još ne znaju sve Sezamove tajne (a ko ih i zna?) zdušno pozdravili Trenutno se uputstva pakuju i šalju teško da bi uskoro trebalo da ih dobiju svi koji su se u međuvremenu učlanili

KOMUNIKACIJE — Dejan Ristanović

Jedna od osobina koja odlikuju rasprave u Sezamovim konferencijama svakako je i nepredvidljivost tema *protokoli* standardno siromašne porukama (toliko da se nekoliko puta postavljalo pitanje njenog opstanka) ovoga meseca je apsolutno dominirala konferencijom KOMUNIKACIJE I to kroz nekoliko paralelnih diskusija maksimalne brzine prenosa podataka sinhronim i asinhronim modemima najpogodniji protokoli za *download* preko paketne mreže (vidi se da je JUPAK poskupeo) aktiviranje i deaktiviranje MNP a na modemima koji ga hardverski podržavaju i na MX5 emulatoru Rasprava je bila izuzetno produktivna bez polemičkih tonova i sa obiljem korisnih informacija

Živo je bilo i u temi *hackers* sve je počelo jednom porukom sa *Usenet*-a U njoj je pomenut neki student sa Univerziteta Ilinois (USA) koji je poslao preteće pismo Klintonu elektronskom poštom u kome se umesto adrese nalazilo *Allmightyy Never Gonna Catch Me* Naravno FBI je stupio u dejstvo i haker je ubrzo pronađen i uhapšen Mnogi naši korisnici su se čudili kako je mogao biti uhvaćen ako se nigde nije koristio svojim imenom što je bila dobra prilika da se opišu neki od mehanizama *Internet*-a i drugih mreža, a onda je diskusija skrenula na pitanje postavljanja „lovca" na domaćim telefonskim linijama i njegove efikasnosti Nekoliko korisnika iznelo je svoja iskustva po tom pitanju njihov zajednički sadržalac jeste da ima raznih tehničkih problema ali da onaj ko uznemirava druge telefonom najzad bude lociran i kažnjen

Temom *modemi* dominirala je rasprava o tonskom biranju tj registru S11 koji određuje brzinu tonskog biranja. Preporučeno je da se u *init* string stavi S11 = 55 što je optimalna vrednost za domaće pošte Tema *fax* donela je *patch* za verziju 3 04 poznatog paketa *WinFaxPro* i diskusiju o „alternativnim" faks programima. OL2FAX je dobro dobio kritike ali je „pao" na pokušaju prelaska sa veze glasom na slanje faksa – kažu da treba otkucati , ali to kod nekih izgleda ne radi Ostaje *WinFax*

Tema *kom.programi* donela je (po ko zna koji put) raspravu o upravljanju udaljenim računarom preko modema i o programu *pcAnywhere/LAN* u temi *procomm* stigla je najava paketa *Procomm Plus for Windows 2.0* a u temi *telemate* bavili smo se izvršavanjem ovoga programa u pozadini i prioritetima u *Windows*-u (preporučeno je da se stavi 50/50 u odgovarajući PIF)

Tema *email* donela je veoma zanimljivu diskusiju o (eventualnom) povezivanju Jugoslavije sa globalnim *Internet*-om korisnici su pokazali toliko interesovanja da je 5 marta održana osnivačka skupština YUCCA, Jugoslovenskog udruženja za računarske komunikacije (*Yugoslav Computer Communication Association* skraćenica YUCCA treba da asocira na onvo životu *Lifelean Lifdee Agavaceae*) detalje o ovom udruženju reći ćete u pročim „Računarima" Na Sazamu je kreirana tema *yucca* u konferenciji EXTRA tamo se prenose poruke iz odgovarajuće grupe *Beonet-a*

U temi *vax* govorili smo o nekim reorganizacijama u RC ETF prenosu datoteka sa VAX-a na VAX preko JUPAK-a (ovaj problem nije rešen trebalo bi napisati novi PSIPAD softver) tema *yu bbs* donela je diskusiju o pokušajima priključenja Intel BBS a na JUPAK i ažuriranja liste BBS-ova u Jugoslaviji dok smo u temi *ptt* pratili pojeftinjenja telefonskih impulsa – najzad stigosmo do 2 pare za impuls Pomenimo i dva zanimljiva priloga u temi *razno* Srđan Pantić (*spantic*) je poslao „Vodič za bezbednost" zvanični dokument YUCCA koji opisuje bezbednost na kompjuterskim sistemima i daje predloge za izbor lozinki i zaštitu od hakerskih (i drugih) napada Sa stranih mreža je prenesen *The USENET Guide to Power Posting*, neka vrsta ironičnog vodiča za one koji žele da „nauče" kako se „diskutuje" u raznim kompjuterskim konferencijama

PC.OS — Pavle Peković

Početkom marta u konferenciji PC OS raspravljali smo o temama koje bi bolje pristajale konferenciji PC UTIL EMS XMS UMB Uz diskusiju je rešeno i nekoliko problema neslaganja QEMM-a i *DoubleSpace*-a kao „podizanje" nekih programa u „gornju" memoriju Kada je o ovome reč nezaobilazan je program VGAMax koji je pravio probleme u radu sa *Telemate*-om ali su problemi brzo otklonjeni Mnogo bure izazvala je vest da *Microsoft* koji je izgubio spor sa firmom *Stac* neće više isporučivati DBLSPACE uz MS DOS korisnike je naročito zanimalo kakve su tačno odredbe te presude ko kome plaća koliku odštetu i da li će sledeće verzije DOS a ponuditi nekakvu kompresiju Pričalo se i o nekim zaboravljenim temama, kao što je istovremeno otvaranje više od 20 fajlova pod DOS om kao i strukture *environment*-a

Windows korisnike zanimali su formati raznih fajlova kao što je RTF (*Rich Text Format*) od koga se prave *Windows Help* fajlovi zatim *Word for Windows* format, i neki drugi Ipak hit tema je *Windows for Workgroups v3 11* koji je pored odušavljenja zbog povećanja brzine rada doneo i mnogobrojne probleme vezane za 32 bitni pristup disku a naročito mrežne probleme Videli smo kako se WFW 3 11 slaže sa drugim mrežnim programima (*Novell* TCP/IP) a bilo je i dilema pri izboru pratećih drajvera

Korisnici *Unix*-a rešavali su problem instalacije modema objašnjavali kako su „DOS novotarije" nešto što na *Unix*-u postoji godinama, pričali o novostima SCO ODT-a verzije 3 0 kao i o „štimovanju" *kernel* parametara Razgovaralo se i o problemima nedavka *Unix* softvera nezaobilazna tema su *public domain* verzije samog *Unix*-a naročito Linux NetBSD FreeBSD

PC PROG — Dejan Jelović

U centru teme *algoritmi* konferencije PC PROG bile su ANSI sekvence ili bolje reći „spasavanje" od njih – kako napisati program koji će ih ukloniti iz nekog LOG fajla. Razgovarali smo i o generatorima slučajnih brojeva koji bi „radili" po određenoj raspodeli što obično ne predstavlja problem uzme se „običan" generator slučajnih brojeva (uniformna raspodela, funkcija *rnd*) i onda se „propusti" kroz inverznu funkciju željene raspodele Jedina muka je što se inverzna funkcija ne može baš uvek naći

Zoran Životić je u temi *ms dos* izneo jednu dosad nepoznatu (?) osobinu DOS a. kada funkciji *findfirst* (0x4E) navedete neki uređaj kao što su CON PRN i slični funkcija javlja da postoji i dodeljuje im atribut 0x40 koji je u Microsoftovoj dokumentaciji naveden kao *reserved for future use* Prvo smo pomislili da je to novitet DOS-a 6 0 ali se pokazalo da stvar važi i na verzijama 5 0 i 3 31

U temi *windows* najveća povika bila je oko *help* kompajlera – koji valjaju koji su komforni itd Vredi pomenuti i Dr Help koji od *Word for Windows* dokumenata pravi RTF fajlove kao i potragu za „dekompajlerom" HELP fajlova koje još uvek nije urodila plodom

Alokacija memorije iz TSR-a mučila je programere u asembleru U temi *jezici* su *dragisha* i *nbalocanin* nastavili svoju maratonsku diskusiju „Clipper protiv ostatka sveta" teško da će iz te priče neko izaći kao pobednik, ali su zato oni koji su je pratili naučili dosta novih stvari

Paskal se ovog meseca ipak pokazao kao akademski jezik glavne diskusije u temi *pascal* vodila se oko množenja matrica, mada su „zakačene" i stvari kao što su pojmovi pretraživanje direktorijuma i interrupti C programeri su se najpre zabavljali generatorima slučajnih brojeva a onda je stigao novi *Borland C++* Iskustva uglavnom iznosena u temi *cpp* prilično su „pomešana" kompajler zaista omogućava brzo i lako pisanje *Windows* aplikacija ali je i veoma zahtevan korisniku koji ima sasvim pristojan 486 računar sa 8 megabajta RAM a je čak i za najmanje programe davao poruku *Out of memory* proradio je lepo tek kada mu je dodat *swap* fajl od 10 mega što zajedno daje zastrašujućih 18 megabajta!

Clipper programeri su mučili svoju muku sa Six drajverom koji je često davao *VM integrity failure* Izgleda da greška nije do Six a već do *Clipper*-a a da je Six samo iznosi na videlo pošto intenzivno koristi memoriju Ah taj *Clipper*!

PC.USER — Nikola Bošković

Sa pojavom programa *Word for Windows 6 0* diskusije u temi *word* je zivnula Naročito smo se bavili (javno dostupnim) *upgrade*-om na verziju 6 0a koji nije bio primenljiv na sve kopije *WinWord*-a rešenje je detaljno opisano u „Bajtovima lične prirode" ukratko u WIN INI datoteci (a i na drugim mestima) treba pronaći red **winword exe**=c 6 0 0 2222 (1033) – c:\winword\setup stf i zameniti 2222 sa 2226 što je oznaka evropske verzije programa koju većina korisnika ima.

Podsetili smo se jednog starog problema vezanog za GEM *Venturu* ako želite da uvezete ASCII tekst stavljajte po jedan prazan red između pasusa U protivnom će *Ventura* shvatiti kompletan tekst kao jedan pasus a pošto je pasus ograničen na 8 kilobajta.

Asocijacija korisnika CAD programa polako poprima izgled prave organizacije Usvojen je i naziv same organizacije (CADUA *CAD Users Association*) i logo Tokom aprila završiće se i pravni deo posla oko registracije udruženja

Korisnici su razmenili informacije o pretplati na strane časopise BYTE PC Magazine i druge više firmi prima pretplate ali se treba dobro raspitati pre nego što se daju pare

PC.UTIL — Bojan Pavković

Glavna tema konferencije PC UTIL bio je na žalost poznati virus Mikelandelo koji „udara" 6 marta.

KAKO POSTATI ČLAN

Postupak učlanjenja [illegible] podesite parametre komunikacije na 2400 8N1 [illegible] MNP modem, uključite MNP i pozovite (011) 648-422. Posle [illegible] poruka Sezam će vas [illegible] ljanja.

Username

Na vama je samo da otkucate NEW (novi korisnik) a Sezam će vas dalje voditi kroz proces prijavljivanja.

Izbor imena

Jedan od prvih podataka koje treba da date je puno ime i prezime a potom i pseudonim pod kojim ćete koristiti usluge Sezama. Jednom izabran pseudonim ostaje vaše trajno ime na Sezamu (ne može se menjati), što znači da ga treba pažljivo izabrati. Predlažemo da se još pre poziva opredelite za pseudonim koji bi trebalo da podseća na vaše ime i prezime, ukoliko nemate ideja. Sezam će vam predložiti pseudonim sastavljen od vašeg prezimena i prvog slova imena. Pseudonim može da ima između 4 i 16 znakova (preporučuje se 5-8 slova, pošto više znakova znači više kucanja, kako za vas tako i za one koji sa vama komuniciraju) i u njemu se mogu naći isključivo slova i znak tačka.

Izbor lozinke

Sledeći korak je izbor lozinke (password) kojom štitite vaš račun. Za razliku od pseudonima koji je javna informacija, lozinku znate samo vi i Sezam. Zato je izaberite pažljivo, naročito izbegavajte da lozinka bude jednaka pseudonimu ili da se sastoji od vašeg imena, nadimka ili imena nekog člana porodice – takve lozinke su lake pogodit, što donosi različite probleme, pre svega onome čija je lozinka ukradena. Izabranu lozinku kasnije možete da menjate.

Izbor radnih parametara

Ostaje još da izaberete radne parametre (arhiver, protokol i slično – ne brinite ako još ne znate šta je šta, pošto će Sezam uvek predložiti uobičajene vrednosti).

Unos ličnih podataka

Pri kom unosa adrese i telefonskog broja budite veoma pažljivi jer netačno unesena adresa onemogućava kontakt Uprave Sezama sa vama, a samim tim i vaše učlanjenje. Unošenjem adrese i ostalih podataka završena je prva faza vašeg prijavljivanja na Sezam.

Radsevno..

Sledećeg radnog dana biće vam poslato pismo sa obaveštenjima o Sezamu i posupnjenom uplatnicom. Ukoliko želite da proverite da li su podaci koje ste uneli ispravni, pozovite sledećeg dana, unesite izabran pseudonim i lozinku i Sezam će potvrditi da vam je pismo poslato ili zahtevati da dopunite podatke. Pošto dobijete pismo, treba da izaberete period na koji se pretplaćujete i na osnovu toga uplatite i uplatite odgovarajući iznos.

Cene za pojedince (60 min dnevno, 30 min po pozivu)

za 2 meseca	10 NDIN
za 5 meseci	20 NDIN
za 8 meseci	30 NDIN
za 12 meseci	40 NDIN
za 18 meseci	50 NDIN
za 35 meseci	100 NDIN

Cene za firme (480 min dnevno, 30 min po pozivu)

za 2 meseca	50 NDIN
za 5 meseci	100 NDIN
za 8 meseci	150 NDIN
za 12 meseci	200 NDIN
za 18 meseci	250 NDIN
za 35 meseci	500 NDIN

Na uplatnici koju dobijate od redakcije upisane su u redu ispod vašeg korisničkog imena i cena u novim dinarima. Cene pretplate možete saznati i na Sezamu, komandom PRETPLATA, a ako se na dan uplate javite upravi glasom na (011) 663-746.

Nakon izvršene uplate fotokopiju potvrde pošaljite pismom na adresu „Računari", Bulevar vojvode Mišića 17, Beograd ili faksom na telefon (011)648-140 ili (011)647-855. Uprava će vam na dan prijema potvrde odobriti probni pristup sistemu, a sledećeg dana da na vašu adresu biti poslato Uputstvo za upotrebu Sezama, atraktivno štampana knjižica pomoću koje ćete upoznati sve mogućnosti sistema.

Prekopreno učlanjenje

Oni kojima se žuri mogu postati korisnici Sezama za svega nekoliko sati. Javite se na opisan način a onda, ne čekajući pismo, izvršite uplatu u pošti i dana navodeći podatke: **Pošiljalac:** vaše ime, prezime i pseudonim koji ste izabrali. **Primalac:** BIGZ – Računari. **Svrha uplate:** Pretplata na Sezam. **Žiro račun:** 40802-603-6-23264, **poziv na broj** 05 108-5. Potvrdu o uplati pošaljite faksom do 14 časova (radnim danom), već u popodnevnim časovima ste punopravni korisnik Sezama!

Pretplata na Sezam se može izvršiti i gotovinski (u BIGZ-u, svakog dana od 8 do 14 časova, ili preko čitavog dana (8-24 časa) na punktu Sezama u knjižari „Plato", Akademski trg broj 1, kod Filozofskog fakulteta).

Dvostruko prijavljivanje

Ukoliko tokom predstavljanja imate problema (npr. veza se prekine) pozovite ponovo i prijavite se navodeći kompletne podatke. Pažnja: ako se prijavite dva ili više puta, koristite ono korisničko ime koje je napisano na uplatnici.

Koliko važi uplatnica

Naša uplatnica važi godinu dana, a pretplatu treba uplatiti po cenama koje **važe na dan uplate**. To ne znači da se morate ponovo prijavljivati – dovoljno je da pozovete Sezam, predstavite se izabranim imenom i lozinkom i sistem će upisati ažurne cene pretplate.

Konferencija	Poruka	Dužina	Datoteka	Dužina
UPUTSTVO	1	1024	0	0
NOVOSTI	726	3288036	18	222380
RAČUNARI	85	55882	0	0
SEZAM	520	338406	6	72613
KOMUNIKACIJE	680	465766	6	254984
PC HARD	400	286137	1	4767
PC OS	741	555547	16	827820
PC PROG	483	382065	8	198215
PC USER	263	174744	6	101755
PC UTIL	288	201672	6	371477
ATARI	94	68326	9	123792
AMIGA	123	95715	13	650088
ORKA	818	848380	21	1281853
IGRE	322	187450	11	874067
FORUM	408	394698	3	28485
CIVILIZACIJA	3406	2978612	18	546482
EXTRA	526	485595	8	170679
CHAT	35	20622	0	0
VICEVI	300	246972	1	31503
MALI OGLASI	1006	663495	1	6040
IMTEL	9	4861	0	0
CET	2	834	0	0
Ukupno	11312	11754979	152	5763880
Privatne poruke	11330	8565096		
Grupe	7167	2778568		

Broj poziva	33482
On line vreme	7320.27:22
Chat vreme	1318.10:26
File transfer vreme	2685.31:32
Upload datoteka	1066
Upload kb	54910
Download datoteka	31243
Download kb	1723648

Upozoranja izgleda nekima nisu pomogla, čak ni pojava nove verzije programa *Scan*, *Clean* i *VShid* (v112) koji su se na Sezamu pojavili prvih dana marta nije sprečila da 6. mart prođe bez uništenja podataka na nekim diskovima. Što je najgore, stradali su i neki koji su znali da imaju Mikelanđela pa su, umesto da ga uklone programom *Clean*, „pomerali" datum i pri tome pogrešili. Prva poruka koja se bavila fatalnim posledicama delovanja ovoga virusa pojavila se još 5. marta kada je jedan korisnik vraćao datum „na normalu" zaboravivši da je ponoć već prošla. Posebna priča su oni koji su pomerili datum ali su zaboravili razliku između „6 sati ujutru" i „6 sati popodne" pa je za računar i u ponedeljak 7. marta pre podne bio u toku zlosrećni 6. mart.

Osnovna ideja većine poruka bio je pokušaj spasavanja onoga što se spasti može. Od pomoći su *uninstall* diskete koje DOS kreira pri instalaciji (čuvate li ih?), pošto je na njima kopija particione tabele i *boot* zapis, na žalost prebrisani podaci sa početka particije (prvih 100 cilindara?) ne mogu se nikako oporaviti. Kako god da se spasavanja podataka završi, svakako proverite sve svoje diskete jer je gotovo sigurno da se Mikelanđelo „uvukao" i na njih – ako ste ove godine stradali, nemojte grešku ponoviti iduće. Za one koji nisu imali prilike (niti će je, nadamo se, imati) da vide delovanje Mikelanđela, recimo da, suprotno nekim verovanjima, nema nikakvih atraktivnih vizuelnih efekata – samo se disk zavrti i podaci nestanu sa njega! Destruktivno delovanje virusa Mikelanđelo bilo je povod za diskusiju o zakonskoj regulativi koja se odnosi na pisanje i širenje virusa. Interesantno je primetiti da ukoliko se na viruse i njihove autore primenjuju uobičajeni zakoni o krivično pravnoj zaštiti imovine, kazni podleže i onaj ko vam je na disketi doneo virus, jer je učinio „delo iz nehata". Pored Mikelanđela, pomenuta su i dva nova (?) virusa. Prvi se zove Žarinov (neki ga zovu „Žirinovski") i razmemo je jednostavan, uklanja se *Clean*-om ili prostim startovanjem NDD-a. Drugi se zove Stamford, instalira se u particionu tabelu i *boot* sektor, program MSAV ga ne detektuje, a program *Clean* nije u stanju da ga eliminiše. Bilo je reči i o novoj verziji programa *Scan* (2.0 beta) koja je značajno brža od prethodnih.

U temi *grafika* diskutovali smo o postojećim programima za animaciju na PC računarima, pre svega o programima *Topas*, *QFX*, *Rio*, *Animator Pro*, *3D Studio*. Konstatovano je da je *Animator Pro* i pored onog „pro" u imenu pogodan jedino za amatersku upotrebu jer se boje biraju iz osmobitne palete. Sve u svemu, ništa bez *3D Studia*.

Bilo je reči i o programima koji pronalaze datoteke (i) u arhivama, emulatoru TIM-a 011 ili baš programu koji je u stanju da čita i piše diskete formatirane za TIM (nismo ga pronašli), problemima sa *DOS Navigator*-om, slanju *init* stringa iz terminal moda, „poremećaju" memorijske konfiguracije QEMM-a koja nastupa ako se greškom obriše LOADHI RF ili DOS UP DAT.

ATARI — Bojan Stanojević

Posle nekoliko meseci čekanja konferencija ATARI je dobila novog moderatora Bojana Stanojevića (*bojs*). To je rezultiralo kako živjom raspravom u konferenciji tako i novim verzijama programa u javnom vlasništvu u ATARI direktorijumima. Tokom februara konferencija se napunila pa je otvorena nova, treća po redu, sa nešto izmenjenim imenom (bez sada već anahronog sufiksa ST) i novim spiskom tema.

U martu je najživlje bilo u temi *obrada.teksta* – noviji korisnici Atarija interesovali su se za pisanje tekstova koji obiluju matematičkim formulama. Predloženo je nekoliko programa, ali izgleda da TeX ostaje bez premca. Živo je bilo i u temi *komunikacije*: pokušali smo da odgovorimo na pitanje postoji li softverska emulacija MNP protokola za *Atari* računare. Posle nekoliko odgovora od „upućenih" korisnika da takvo nešto ne postoji, našao se i program koji obezbeđuje softverski MNP, doduše u demo verziji koja je vezana za prilično primitivni terminal emulator. Prva iskustva pokazala su da pouzdanost nije na naročito visokom nivou. Možda bude bolje u produkcionoj verziji koja će navodno sarađivati sa većinom postojećih komunikacionih programa.

Večita tema „kako ubrzati računar" nije mimoišla ni ATARI konferenciju: uz zamenu procesora 16 MHz verzijom možete udvostručiti radni takt na ST mašinama, ali samo interno (slično kao kod STE mašina). Preneseno je uputstvo iz nemačkog časopisa, a bilo je i saveta oko izbora dodatnih čipova, kao i o tome koliko se ceo posao isplati.

Programerski (hakerski?) kutak bio je ispunjen diskusijom oko toga da li se može pročitati listing programa u GFA basic-u koji je snimljen sa auto startom. Za sada preovlađuje mišljenje da nije moguće s obzirom da se u program tada ne ugrađuju imena promenljivih.

Stara tema kako razmeniti podatke između Atari je i PC-ja dobija nove momente prema svedočenju nekih korisnika MS DOS 6.0 bez problema čita (gotovo) sve Atarijeve formate

ORKA — Slobodan Kalazić

Mart je u konferenciji ORKA bio izuzetno živ: gotovo 1000 poruka sa ukupno 829728 bajta teksta! Tema *najbolji* ponovo je pokazala kako tehnike napredu je računari su sve bolji pa se i argumenti menjaju, ali nekako više kvantitativno nego kvalitativno, pri čemu odnos i pozicije ostaju otprilike isti. I dalje se suočena strana ne shvataju i dalje ne mogu da se slože oko dosta stvari, ali ima i stavova oko kojih su (gotovo) svi saglasni. Radnja promena načina vođenja diskusija do koje je došlo za ovih nekoliko godina: današnji diskutanti ispoljavaju daleko manje netrpeljivosti, agresivnosti i neprijatnosti prema neistomišljenicima.

U temi *mac* nastavljena je diskusija o hardverskim i softverskim PC emulatorima na Makintošu. Na to se nadovezao razgovor o umrežavanju računara ove dve platforme, tj. kako se PC može umrežiti sa Mekovima koji su već povezani u *LocalTalk* ili *EtherTalk* mreži. Nastavljena je i diskusija o kompatibilnosti datoteka iz aplikacija koje postoje i za Meka i za *Windows* (*Quark, FrameMaker, Word, Excel*...). Bilo je govora i o PICT formatu slika na Makintošu i razlikama između PICT 1 i PICT 2 formata. Polovinom marta smo dočekali vesti o pojavi *PowerPC* Makintoša.

U temi *electron* se odvijala uobičajena razmena informacija i potražnja komponenti. Pošto smo nedavno dobili PD asembler za Intelove mikrokontrolere iz porodice 8051, sada su neki Sezamovci tražili i C, Pascal i Basic kompajlere jer su čuli da i ovi postoje.

Klub programera, najstariji domaći Klub ove vrste, koji je nedavno proslavio 11 godina postojanja, konačno je našao pogodnu prostoriju za sastanke svakoga petka u 20 časova na uglu ulica Vele Nigrinove i Jovana Rajića na Crvenom krstu. Utisci o osoblju i usluzi bili su više nego povoljni, tako da je *jerry* koji je našao prostoriju zaslužio „Orden zasluga za Klub" koji mu je (nekoliko puta) uručila ana. Od zanimljivih događaja iz Kluba pomenimo i dve torte koje su Sezamovci uspešno „disasembilirali" poslednjeg petka u martu uz rođendanske čestitke Danici i CDraganu.

IGRE — Dragan Čitaković

Tokom marta u konferenciju IGRE stiglo je nešto manje poruka nego proteklih meseci. Ipak uspešno je odgovoreno na većinu postavljenih pitanja. Najviše poruka u temi *arkadne* bilo je vezano za pojavu serije EPIC-ovih flipera pod nazivom *Pinball Fantasies*. Mišljenja igrača su podeljena: neki su oduševljeni, a drugi tvrde da je *Tristan* neprevaziđen. U svakom slučaju, dobra zabava je zagarantovana. Raspravljali smo i o preprekama u igrama *Syndicate* i *Fury of the Furries*. U temi *avanture* je vladalo malo zatišje, ali se uz poruke onih koji su tek sad otkrili *Secret of Monkey Island* našlo i dosta poruka vezanih za avanturu *Hand of Fate*. Pojava dugo očekivane nove verzije *Simcity 2000* „razbudila" je mnoge Sezamovce: bio je zamerki na trodimenzionalni prikaz i na stroge hardverske zahteve, ali je za mnoge „graditelje gradova" *Simcity 2000* bio prava stvar. Dobili smo i opis jedne od prvih igara koje se isporučuju isključivo na CD-ROM-u, *Rebel Assault*. Oni koji su videli kažu da igra, osim mnogo grafike i zvuka, ne donosi nešto posebno novo – uludo (?) potrošenih 400 megabajta. Ipak, vredi pogledati!

Što se teme *strategije* tiče, ovog meseca nije bilo iznenađenja ni najava novih igara. I dalje su prisutne rasprave o taktikama za stvaranje civilizacija, kako obična tako i svemirske, a podsetili smo se i na caka koje olakšavaju savladavanje prepreka u igri *Dune*. U temi *frp* dalje se govori o igri *Lands of Lore*, a priličnog interesovanja izazvala je i pojava nove Microproseove RPG igre *Bloodnet*. Radnja se odigrava u budućnosti, a cilj je da uz pomoć pet likova spasete glavnog junaka kojeg su izujedali vampiri! Što se logičkih igara tiče, Saša Pršica je ispunio obećanje i poslao poboljšanu verziju igre *Mindform* sa ugrađenim zvukom i dodatnim poteškoćama pri igri. Stigao je i program *Piano Fantasy* istog autora – uz njega ćete komponovati trokanalnu muziku na PC zvučniku!

FORUM — Bojan Tepavčević

Posle pozamašne pauze, tokom marta su oživele političke teme u FORUM-u. Povod je bio tragičan slučaj samoubistva ćerke generala Ratka Mladića, koji je u prvi mah inicirao raspravu o motivima za takav čin, da bi kasnije došlo do žestoke diskusije po pitanju rata u Bosni uopšte. Posle intenzivnih „artiljerijskih" duela krajem meseca došlo je do izvesnog smirivanja situacije na forumskom frontu, uz povremene provokacije „zaraćenih strana". Zanimljivo je da u FORUM-u i dalje nema previše poruka koje se bave ekonomskim programom, mada su nekoliko puta pomenuti problemi u bankama sa zamenom novca i napravljena paralela sa sličnim „tehničkim problemima" sa kraja 1990. godine.

I u temi *iseljenje* je bilo zanimljivih priloga. Diskutovalo se o ulaznim vizama za Australiju, Novi Zeland i neke druge države. Posebno ističemo prilog Aleksandra Radovanovića (trenutno u Južnoj Africi) koji je poslao detaljan opis forme i sadržaja autobiografije (*Curriculum Vitae*) kakvu na Zapadu treba podneti pri konkurisanju za posao.

U temi *devojke* i dalje traje permanentna tribina o odnosima među polovima, koje nerelko bude začinjena i ponekom GIF ilustracijom.

Sredinom marta konferencija FORUM 10 je dostigla tehnički limit od 5000 poruka i tako otišla u istoriju, tj. u *read only* status. Za nove poruke otvoren je jedanaesti tom FORUM-a.

CIVILIZACIJA — Dejan Balinda

Odavno više nije iznenađujuće da konferencija CIVILIZACIJA bude vodeća po broju poruka. Međutim, ostaće zabeleženo da je tokom marta u ovu konferenciju pristiglo 3408 poruka, što predstavlja apsolutni i teško dostižni rekord. Najživlje je bilo u temi *o jeziku* gde se sa neočekivanom strašću za konferenciju ovakvog naziva vodila žustra polemika o odnosu ćirilice i latinice kod nas. Čudno je da su ljudi koji bez ikakve sumnje poznaju i koriste oba pisma baš toliko pristrasni: pitanje je da li su ovome kumovale trenutne političke okolnosti?

Tema *filozofija* počastvovana je kompetentnom raspravom o odnosu matematike i filozofije. I među nama malo od rečenog razumeli, a možda i upravo zbog toga ostao je utisak da je iznesano nešto veoma važno. *Istorija* se bavila seobom Srba, a *nauka* tunel efektom i Hajzenbergovim principom neodređenosti. U temi *književnost* odavno nije bilo tako divnih priloga kao što su ovog meseca bili oni u kojima je zablistao Danil Harms. Tema *film* donela je nastavak filmografija koje su i ranijih meseci bile veoma zapažene. Dobili smo pismo iz Toronta „Jill" u Vatikanu čudili se kako treba disati u Sikstinskoj kapeli.

U temi *škola* smo analizirali rezultate i domete štrajka srednjoškolaca, a aforizmi Stanislava Jerzi Leca pokazali su nam da se najlepše stvari često nalaze u malom pakovanju. Sukobili smo i svoje stavove u pogledu moralne, ali i svake druge odgovornosti naučnika za rezultate svoga rada, sa posebnim osvrtom na nuklearna istraživanja.

U jednom ovako kratkom biltenu nije moguće čak ni nabrojati sva dragocena poruke, s obzirom da je svakog dana tokom meseca marta pristizalo više od stotinu osvrta na čudesni svet koji nas okružuje i na odnose koje smo među sobom izgradili.

VICEVI — Darko Staničić

I tokom marta na Sezamu je bilo dosta novih i zanimljivih viceva, uz neizbežno ponavljanje ponekog starog. Tema *english* se unekoliko smirila, premda smo u njoj pronašli jedan test pomoću kojeg svako može da ispita svoje puntanstvo (odgovorite na 100 „škakljivih" pitanja i tako „izračunajte" svoj „procenat puntanstva") a i zanimljivu pričicu o tome kako bi avionski saobraćaj izgledao kada bi autori operativnih sistema vodili avio kompanije. Recimo „Kompanija Windows": Aerodromski terminal lep i šaren sa ljubaznim stjuardesama, lakim prilazom avionu, jednostavnim poletanjem... iznenada u vazduhu avion eksplodira bez ikakvog upozorenja.

U ostalim temama pominjali smo Mujo i Hasu, plavuše, dizalaše, političare. Pretresli smo i huligane raznih zanimanja i nacionalnosti. I dobili čitavu kompilaciju „nastašluka". Izbrali smo, naravno, i vic februara koji je poslao vitez *koja*: Dolazi baba na groblje i pita grobara: „Izvini, sinko, gde je ovde parcela 375?" „Me lezi baba, da očeš?"

Ako ste ikada razmišljali o tome zašto je pivo bolja od žene, na Sezamu možete naći još 103 razloga u prilog pivu! Za kraj, kao i obično, jedan od viceva koji su obeležili mesec: Šta je to jnpa? Otvoj u zidu, spojava za uvajčavanje. JUPAK bez KI A šta je to EJ-JOJ? Jupa u pjoglamu.

Nastavak sa strane 63

definiše identifikator VPI (*Virtual Path Identifier*). Takođe grupa VPC-a se prenosi i komutira zajedno. Slika 9 prikazuje mrežu sa tri VP kanala između A i B. Računar A ove veze vidi kao VPI je p, q i r, a B kao p2, q2 i r2. Slično kao i kod VCI-ja komutacioni čvorovi T1, T2, T3 definišu međuvrednosti i grade tabele zamene. VCI se ovde ne menja tako da A može da ostvari nekoliko VC veza preko bilo koje od VP veza.

Sloj iznad ATM sloja naziva se ATM adaptacioni sloj (AAL – *ATM Adaptation Layer*). Kako mu i govori svrha sloja je da podatke iz raznih izvora (datoteke, uzorci glasa, slike ili video) pretvori u ćelije koje se šalju ATM sloju. Za ovaj sloj je definisano nekoliko klasa servisa u zavisnosti od podataka koji se prenose. Na primer, pri prenosu video signala od ćelija se zahteva da podatke prenosi konstantnom brzinom i bez kašnjenja. U zavisnosti od zahteva za ovaj sloj je definisano niz protokola.

BITOVI U ŽICI

Za prenos podataka na veće razdaljine ATM ćelije se konvertuju u SONET format. SONET (*Synchronous Optical NETwork*) deo je niz standarda poznatih pod imenom SDH (*Synchronous Digital Hierarchy*) donetih od strane CCITT-a. Po SONET standardu osnovna brzina prenosa je 51.84 Mbps i naziva se STS-1 (*Synchoneus Transport Signal level 1*) ili OC-1 (*Optical Carrier level 1*). Oba nivoa (OC i STS) praktično opisuju niz bitova. OC opisuje niz bitova definisan STS standardom posle konverzije u optički signal (Tabela 1). Bitovi sa SONET-om prenose upakovani u okvire. Okvir je matrica bajtova sastavljena od 90 kolona i 9 redova. Za OC-n prenosiće se n okvira, odnosno OC 1 prenosi podatke okvir po okvir, dok OC 3 prenosi grupe od po 3 okvira.

Brzina Prenosa (Mbps)	STS/OC
51 84	1
155 52	3
622 08	12
1244 16	24
2488 32	48

ATM ćelije se „pakuju" u SONET okvir redom kako nailaze sve dok ne popune red. Kako 90 nije deljivo sa 53, jedna ćelija se može naći delom na kraju jednog, a delom na početku sledećeg reda.

VIZIJA BUDUĆNOSTI

U razvoj telekomunikacionih sistema širom sveta uloženo je mnogo godina i novca. Nemoguće je preko noći postaviti hiljade kilometara kablova i promeniti odgovarajuću infrastrukturu. Neke zemlje, SAD pre svega, svojim projektom *information highway*-a uveliko rade na primeni nove tehnologije. Već je moguće nabaviti svu potrebnu opremu za izgradnju lokalnih mreža baziranih na ATM-u. Procenjuje se ipak da za većinu korisnika u svetu nove tehnologije neće biti dostupne u širim razmerama pre 2005. godine. Kakva će zaista biti uloga ATM-a, ostaje budućnosti da kaže.

Sto brojeva „Računara" predstavljaju pravu riznicu znanja i informacija: za ovih deset godina objavili smo 100 brojeva (99 redovnih i jedno specijalno izdanje) i u njima 2719 tekstova iz pera 314 autora.

HARDVER

KOMPONENTE SISTEMA

Mikroprocesori

DEC Alpha - 95/36 96/16 97/7
Intel – 8086 17/26 80486 54/8 70/13 77/12 77/10 Pentium 86/22 93/26 94/14 97/12 klonovi 94/8 96/57 97/37 97/36 99/14 Microway i860/33 76/8 i860XP 82/34
PowerPC – 88/8, 84/34 96/12
RISC – 20/20 31/12 32/22 46/16 48/15 69/26 99/17
Trendovi i tržište – 38/16, 39/15, 45/20 47/15 53/30 72/6 99/12
Ostali – 11/17 13/19 37/14 63/18 16/64, 12/67 22/11 35/16 79/8 HD64180 37/17 68000 16/22 AM 29000 49/12
Poređenje – 52/9 63/6 84/44 93/22 53/14 80386/68020 32/12 68040/80486/80860 51/8 debata 40/17

Koprocesori

8087 – 29/20
80287 – 35/20
80387 – 38/18
Poređenje – 66/28 68/18 69/22 69/20 70/12

Magistrale

AD/DA kartica – 56/17
Dnevnik jedne veze – 29/24
Keš – 43/13
Local bus – 88/22 88/26 89/15
Organizacija – Š/6 79/23
Podešavanje sabirnice – 87/80

Razno

Roboti – 35/75 36/59 37/64 38/60 39/62 43/8 62/30
U svetu komponenti – 11/60 12/62, 13/43 14/63 16/64 17/60, 18/60 19/64 20/65 21/63 23/62, 24/63 25/49 26/61 27/53 31/50 32/25

MREŽE

Commodore – 11/41
Eternet – 73/44
Galaksija – 21/16
Heterogena mreža – 95/31
Mrežni dizajn – 64/33 91/7 Schneider & Koch 85/16 protokoli 66/24 67/24 OSI model 68/38 69/50 server 70/38 stanice 71/38 problemi 72/43 adapteri 72/40 virusi 73/46
Sve mreže sveta – 65/56
Token Ring – 74/38 80/46
TopNet/TopWare – 58/25
Ultra – 63/15
Prikaz opreme

Periferijska oprema

Opšte

Opšte 1/23 Š/8

Grafičke kartice

Opšte – 81/18 91/16
Dvostruki ekran – 73/69
Hercules – 54/10 InColor 28/12
VGA 38/10 55/18, 56/24 67/22 76/16 79/18 80/18, 87/14
Wyse – 700 35/15 900 64/18

Masovna memorija

CD ROM – 68/18 85/18 96/28 99/20 95/42
Disk kontroleri – 78/20 66/18 90/18, 94/62 65/48
Flopečki diskovi – 96/21
Hard diskovi – 67/20 70/14 73/10 74/10, 78/12 80/12, 87/12 91/28

Modemi

Opšte – 9/17 40/32 48/52
2400 bps – Discovery 57/36 ViVe 72/14 53/12 90/22
Faks modem – 78/11 82/18 84/42

REGISTAR NOVE GENERACIJE

2719 tekstova objavljenih u prvih 100 brojeva „Računara" prostire se na 7011 strana, a objavljeno je još 2217 strana komercijalnih informacija. Naslagani jedan na drugi, svi brojevi našeg časopisa predstavljaju impozantnu gomilu visoku oko 55 santimetara i tešku 25 kilograma. Količinu objavljenog teksta teško je proceniti – „Računari" se standardno štampaju sitnijim slovima od većine ostalih časopisa, a naši specijalni dodaci bi, štampani uobičajenim fontovima i sa uobičajenim marginama, predstavljali prave knjige. Ipak, verujemo da nećemo mnogo pogrešiti ako kažemo da je u prvih 100 brojeva „Računara" objavljeno između 80 i 100 megabajta teksta praćenog ogromnim brojem listinga, tehničkih crteža, slika i drugih ilustracija.

Pronalaženje željenog teksta u tom obilju bilo bi tek malo lakši problem od pronalaženja igle u plastu sena, kada se „Računari" ne bi trudili da sistematizuju informacije – jednom godišnje objavljujemo registar tema u okviru koga su tekstovi grupisani po tematici, prezimenu autora, kompjuteru kome su posvećeni. Za ovaj praznični broj pripremili smo registar tema nove generacije koji će vam omogućiti da praktično jednim pogledom pronađete tekst koji vam je potreban. Protekło vreme nam čini da naslovi pojedinih tekstova prestanu da budu bitni – sećamo se jedino čime se tekst bavio. Ukoliko, na primer, želite da pronađete prikaz *WinWord*-a 2.0, pogledajte pod Softver / Prikaz programa / Tekst procesori / *Word* i nađite verziju 2.0: 80/27 znači da se tekst nalazi na strani 27 „Računara 80". Ukoliko umesto broja časopisa piše slovo Š, tekst potražite u specijalnom izdanju „Računari u vašoj školi".

Organizacija registra je strogo hijerarhijska: tekstovi su podeljeni na 'hardver', 'softver', 'programiranje', 'informacije' i 'dodatke', a onda je svaka od tih kategorija dalje podeljena na grupe i podgrupe.

Ostalo

Digitajzer – 11/18 31/14
Film recorder – 85/15
Generisanje zvuka – 73/14 94/52
Miš – 21/18
Monitori – 8/17 86/18 89/16 Here Superstation 3D 89/13 MultiSync 35/12
Ploter – 15/12 43/17 44/14 46/18 81/16 95/18
Trackball – 74/14
Red box – 25/18
Skaneri – HP 78/11 81/14
Strimer – 64/36 69/44
Svetlosna pera – 7/17
UPS – 65/19 80/42
Ostalo Roland Camm 75/14 Roland Stika 73/13

Računari

Džepni i portabl

HP – 41C 33/13 71B 13/22 28C 33/12 95LX 79/8 ostali 21/34 37/10 47/8
Psion Organizer – 36/10 40/16 92/37
Šarp – 80/16
TI – 35/60
Z88 vs TRS100 – 25/8

Kućni

Acorn
Arhimed – 30/10 33/8 33/10
BBC – B 2/23 Master 13/8 Master Compact 20/12
Elektron – 2/21

Amstrad
Amstrad protiv Komodore – 4/21
CPC 464 – 3/10
PC – 20/3 30/12
PCW – 8256 19/8 9512 15/5

Atari
800 – 4/18
PC – 25/16 30/8
ST 520 8/8 12/8 1040 13/24 16/8 mega 52/11 modeli 48/12 58/38
TT – 45/8

Commodore
Amiga – 9/8 9/8 26/17
C64 – 2/27 29/8
C128 – 7/10
Sve Komodorove funkcije – 21/56

Sinclair
QL – 2/18 14/8
Spectrum – plus 3/14, 128 10/11, 15/13 17/11 plus 2 20/4 plus 3 28/6
Sve spectrumove funkcije – 18/54

Ostali
Galaksija plus – 8/6
Izbor računara – 1/8
Macintosh – Mac 2 31/8 PowerMac 98/11
MSX – 6/10
Sve Šarpove funkcije – 19/60
TIM 011 – 33/58 34/60 35/7

XT i AT

Atari PC – 25/18 30/8
Baby AT – 39/9
Compaq – SLT/286 63/12 LTE286 66/12
Hakerski vodič Minhena – 16/12, 17/18
Headland – 71/14
IBM PC – Š/81
Klonovi 21/30 24/8 38/21
ICL M45 – 92/12 92/13
Lire – 47/10
Mitsubishi MP286L – 58/9
Portabl – 57/7 67/4 91/26 Nylap 72/10, Acros 325SE 89/56
Sve IBMove funkcije – 23/60
Šutić – AT SC2016 65/14 286/25 75/12
Tandy – 23/8
Ultra – 63/15

386

Compaq – DeskPro 22/8 386SX 63/10 386/33 64/4
Daewoo – 80/14
Epson – EL3/33 78/6
Olivetti – M386 52/8
PS/2 – porodica 27/9 model 30 29/10 model 50 32/10 model 60 35/66 model 70 58/8 model 80 35/10
Victor – 51/10
Ostali – 71/8 81/22 84/12 85/12

486

Altos – 62/14 64/11
Compaq – SystemPro LT 89/22
Game – 73/4 83/7
Notebook 486DLC – 96/11
Sprint 486 – 87/13
Šutić – 486/50 83/10 486/50 EISA 84/38
Vectra – 56/5
Ostali – 63/7 68/16 73/6 75/8 87/12

Pentium

Modeli – 97/12
MR P/6 – 98/9

Štampači

Laserski

HP – LJ SII 9/20 30/18 41/12 LJ III 69/10 LJ 4 87/31 89/14 LJ 4L 94/10 LJ 4P 97/20
LaserPort PS600 – 66/12
LaserMaster – 89/13 800 81/22
MT 82/14

Matrični

Citizen – 180D 65/23
Epson LX – 800 51/12 850 57/11
Epson LQ – 28/10 83/19 1500 5/20 1040 63/18 870/870 78/19
Facit – B2150 59/12 P6060 60/18 B2450 81/18
Mannesmann Tally – 81 54/17 222 55/10 131/8 65/24 906 71/10
NEC – P2200 80/14
Panasonic – KXP1092 16/10 KXP1540 81/12
Pregled – 2/32 15/8 17/14 32/16
Star – ZA250 84/29 SJ144 86/15

Ostali – MPS 1200 38/13 Philips 72/12 Genipnnt 16 74/12 AEG LP60 84/12 Canon PW1080A 7/7

Ostali

Canon ink jet – BJ130 37/8 BJ10E 83/14
HP ink jet – 18/18 24/13 77/18 500 78/21, 500C 78/10
Kolor – 33/17 98/12
Sa ricellom – 19/8

SAMOGRADNJE

Dodaci

Firmware – 34/22
Spectrum – Devpak iz ROMa 17/26 GENS iz ROMa 9/42 ROM disk 6/60 7/78 Turbodrajv 16/63 17/8 18/61 19/62 20/60 21/60 22/60
Ubrzavanje PCja – 43/20

Računari

Galaksija – 1/50 1/65 3/81 3/80, 4/30 2/80 2/88, generator tona 4/78 fina grafika 5/64 6/64
Optime 64/14 65/7 70/9 71/12
Sklapanje PCja – 47/64
TIM 011 – 5/39 37/12 39/51 40/70 41/15 42/50 45/59 46/08

Ostalo

Brisač EPROM-a – 7/79
Interfejs za štampač – 15/55
Programator EPROMa – 2/82 3/78, 7/32
Simulator EPROMa – 58/74

SOFTVER

KOMUNIKACIJE

Modemi – instaliranje 56/32 komunikacija 57/64 praktikum 62/34 softver 71/34
Domaći sistemi – JUPAK 58/32 73/46 Beointernet 99/46 VIK 50/34 ZBBS 53/10
Elektronska pošta – VAX 66/43 67/28 69/47 71/40 74/42 sa svetom 85/41 90/54 97/54 98/46
Modemske komunikacije – 99/0
Sezam – najave 54/75 upotreba 55/31 56/35, 77/38 komercijalni 58/28 63/46 64/61 66/5 rođendani 77/66 87/61 96/9 97/76 Olimpijada 65/76 66/4 67/8 67/6 tanjug 77/67 elektronska škola 79/6
strani sistemi – sve mreže sveta 80/44 Bitnet 81/42, 82/38 Trickle 76/38 Internet 98/48 Usenet 87/54 98/46 BIX 60/24 61/23 BBS 57/32 59/28 FIDO net 56/34

PRIKAZ PROGRAMA

Baze podataka

Microsoft Access – 93/38
Clanon – 77/34
Clipper – 5.01 73/21 Nanforum Toolkit 60/38, O CLIP 82/30 Blinker 30/32 69/36 5.2 93/35 Six drajveri 98/35 99/36
DataPerfect – 74/34
dBASE – 48/32
dBXL – 49/26
dClip – 3.0 85/46
Force – 72/26
FoxPro – 1.02 74/28 2.0 85/24
Paradox – 4.0 86/28 Windows 86/30, 90/26, 92/40
Paradox Engine – 64/30
IZE – 50/22
Science Adventure – 98/43
Šahovske – NICBASE 86/46, Chess Assistent 90/37 58/40
Wizdom – 93/42

Crtanje i projektovanje

Adobe Acrobat – 96/30
Adobe Photoshop – 93/45
Arts & Letters – 53/22 57/22
AutoDesk 3D Studio – 75/15 2.0 85/28 3.0 98/28
AutoDesk Animator Pro – 78/27
AutoDesk Autocad – 74/18 12 86/33 95/11
AutoDesk AutoCad Architectural – 76/20
AutoDesk AutoCad Solid Modeler – 76/18
AutoDesk AutoShade – 2 79/32
AutoDesk Chaos – 84/14
AutoDesk Renderman – 60/24
Corel – 4.0 94/48
DesignCad 3D – 70/28
Designer – 4.0 96/24
Grasp – 43/31 3.1 46/28
Harvard Graphics for Windows – 2.0 96/24
HiWire – 44/25
Lotus Freelance – 72/30 for Windows 98/39
Mass 11 Draw – 45/15
Micro Cap III – 74/22
OrCAD – 47/24
qCAD – 79/48
smARTWORK – 43/28
SLED – 2.1 54/23
Uporedni test 8 programa – 98/22

DTP paketi

Bitstream Fontware – 51/18
Fleet Street Editor – 46/22
FontGen – V 55/12
FontMonger 87/46
Fontographer – 90/40
SoftCraft – fontovi 45/18, font editor 46/24
TeX – 27/56
Tipotek – 86/44 2.01 93/51
Uporedni test – 92/19
Ventura Publisher – 2.0 50/17 3.0 66/9 4.0 80/30 4.1 86/42

Editori

AWK – 66/52
Brief 3.1 75/40
MicroEMACS – 81/46
Mult-Edit – 91/42
OEdit – 75/42
VDE – 78/36

Igre

ChessMaster 3000 – 87/52 95/62
CoreWar – 34/28 35/41 40/60
Elite – 4/34 12/58 13/64 14/64 17/64
Flight Simulator – 3/73 4/72 2.0 41/78 3.0 51/79
Razbarušeni sprajtovi – 5/60 8/57 7/58 12/56 13/55 14/56 15/65 16/65 17/65 18/56 19/65, 20/66 21/63 22/66 23/63 24/64 25/63 26/62 27/62 28/63 29/60 30/62 31/62 32/62 33/62 34/62 35/76 36/73 37/73 38/73 39/73 40/73 41/65 42/64 43/62 44/93 45/77 46/73 47/76 48/78 49/90 50/92 51/79 52/65 53/76 54/92

Matematički softver

Eureka – 49/72
Derive – 50/20
Gauss – 2.0 79/42
MathCAD – 47/51 3.0 84/24
Mathematica – 75/36 81/40
Math Lab – 68/32, 69/32
Microstat II – 44/22
muMATH – 40/24
Origin 2.24 – 87/37
StatGraphics – 42/26
Sygraph 1.0 – 49/30
Systat 4.0 – 48/21

Operativni sistemi

MS DOS – 3.30 39/20 4.0 47/6 5.0 72/21, 6.0 87/21 81/32 6.2 97/29
DR DOS – 5.0 71/32
Windows – 2.1 63/24 3.0 63/26 67/9 3.1 77/22 77/79 82/18 W4EE 87/28 Win4WG 89/48 81/51 NT 79/28 87/22 98/16 Čikago 87/22
OS/2 – 1.x 52/18 59/6 2.0 81/8 83/22 83/26 84/22
Windows vs OS/2 88/28 90/4
Unix – 62/20 Coherent 65/32 96/45 Interactive 74/18
Novel – 3.11 69/36 73/42, 4.0 95/14, Lite 86/36
Mrežni – DLink 84/18 Lan Manager 70/17 71/36 TopNet 58/25

Programski jezici

Asembler – MASM 39/25 75/18 Optasm 43/26
Borland C – 34/20 47/10 C++ 2.0 65/34 72/51 73/18 TC3.0 86/23 BC3.0 82/22 BC3.1 96/36
Borland Turbo Debugger – 91/40
Borland paskal – 23/56 4.0 37/23 5.0 47/22, 5.5 54/20 6.0 66/22 TP4WIN 72/22 BP7.0 88/31
ObjectVision – 71/28 73/30 2.1 87/42
Hisoft C – 1.1 44/69
JPI Modula 2 – v2. 75/30
Logitech Modula 2 – 40/27 41/23
Microsoft QuickBasic – 4.0 35/32 4.5 52/22
Microsoft Visual Basic – 81/32 2.0 93.24 93/48, 3.0 97/36
Microsoft C – 5.1 53/14, 6.0 64/19 70/24 7.0 90/29 85/20 95/22
Microsoft Quick C – 81/37
Microsoft Visual C++ – 1.0 96/39
Microsoft CodeView – 36/32
Microsoft Fortran – 5.1 82/27
Microsoft QuickPascal – 1.0 55/14
Paskal III Modula – 52/19 54/77, 55/21
Watcom C – 78/30
Zortech C++ – 3.0 90/32
Ostali – Actor 42/15 D 46/24 Diamond 43/24 Forth 43/22 Pro C 55/18 Prolog 46/37 Smalltalk 2.0 87/33 Trilogy 53/19 Views Designer 73/35

Radne tabele (spreadsheets)

Excel – 48/18 3.0 70/20 71/20 4.0 83/30
Lotus 123 – 2.3 76/22 3.1 69/34 3.4 96/38 dodaci 77/28
Lotus Improv – 2.0 99/35
Quattro – 65/28 3.0 74/24 4.0 84/17 86/36 88/36, 5.0 97/32 98/19
Symphony – 2.2 71/24 3.0 87/36
Uporedni test – 81/24

Tekst procesori

AMI PRO – 3.0 89/40
Grammatic – 5.0 92/45 87/42
Microsoft Publisher – 79/38
Microsoft Word – 3.0 20/18 WinWord 1.1 69/30; WinWord 2.0 80/27 WinWord 6.0 99/24
Multi Lingual Scribe – 44/19
Webster – 90/29
WordPerfect – 5.0 40/12 41/18 55/57 5.1 59/20, 60/20 WP4WIN 1.0 79/34 6.0 96/35
WordPerfect Office – 72/36
LetterPerfect – 66/25
PerfectPal 43/32
WordStar – 5.5 57/26, 2000 58/18
XyWrite – 42/18

Uslužni programi

386MAX – 67/12 6.0 81/30
4DOS – 66/35 53/36
ARJ – 73/32 2.41 94/36
BeebDOS – 43/34
CED – 40/26
CPYAT2PC – 44/31
DesqView – 87/14
FastBack Plus – 72/34
File Rescue Plus – 51/17
GrafPlus – 46/21
HUeek – 71/31 90/31
Konverzije grafičkih formata – 96/44
Mace Utilities – 42/22
Norton Guide – 51/22
Norton Utilities – 4.50 48/26 62/18 5.0 65/25 6.0 77/40 for Windows 2.0 85/36
OptTech Sort – 46/26
PC File+ – 58/16
PC Kwik Power Pack – 66/22
PC Tools 4.20 40/21 4.30 44/78 5.10 48/20 50/24 62/18 6.0 66/19 8.0 91/34 for Windows 93/50
PKZIP – 73/32 2.0 89/52
QEMM – 5.0 62/28 5.11 68/34 6.0 80/36. 7.0 95/26
SHEZ – 83/34
Stacker – 89/62 93/31
Vastscreen – 44/30

Ostalo

Ekspertni sistemi – 36/26 29/18 60/30, 63/41
Komunikacioni programi – 71/34
Microsoft Works – 57/30
MIDI – 46/32
PlanPerfect – 5.1 87/18
Recognita Plus – 1.2 75/22
SKY Atlas – 38/14
WinFax Pro – 3.0 95/44

UPOTREBA PROGRAMA

Bajtovi lične prirode

Bajtovi godine – 56/54 67/62 78/56 86/78 98/64
Bajtovi lične prirode – 40/53 41/61 42/57 43/53 44/86 45/69 46/60 47/70 48/70 49/79 50/66 51/70 52/61 53/64 54/90 55/56 58/72 59/66 61/60 63/35 64/65 65/63 66/63 68/64 69/43 70/58 71/70 72/46 73/70, 74/72 75/67 76/52 77/54 79/69 80/98 81/60 82/52 83/54, 84/64 85/60 86/61 87/110 89/64 90/76 91/78 92/76 93/74 94/67 95/65 96/76 97/74 99/66

Baze podataka

Baze na Cu – 76/28
Clipper – prozori 69/48 proširenja 68/58 korisnički interfejs 69/69 help i demo 70/48 78/50 slike 70/52 kalkulator 71/62 adaptacija 74/58 preview 75/

Operativni sistemi

CP/M

MS DOS

Unix

VMS

Windows

Savetnik

Radne tabele

Tekst procesori

Ostalo

PROGRAMIRANJE

ASEMBLERSKO PROGRAMIRANJE

Kućni računari

PC računari

PROGRAMIRANJE ŠTAMPAČA

PROGRAMSKI JEZICI

Bejzik

C

C++

Paskal

Ostali

REKREATIVNO PROGRAMIRANJE

95/68 96/72 97/52, 98/70 99/76
Softverski perpetuum mobile – 12/48
RUN kao LIST – 1/61

TEHNIKE PROGRAMIRANJA

Grafičke tehnike

Animacija – 27/50 28/58 35/46
Fill – 48/52 12/37
Fraktali – 31/60 52/28 65/72
Krugovi i elipse – 24/56 26/49 32/48 34/39 35/48 36/54 37/58
Linije – 41/59 54/61 55/46 56/45 78/46
Modeliranje – 74/70 75/44 79/48 80/55
Opšte – 43/45 52/48

Kućni računari

Acorn BBC – nedokumentovane naredbe 3/32
Commodore 64 – nove azbuke 4/62 zvuk 6/54 nove naredbe 6/31 turbo tape 11/32 sprajtovi 5/56
Spectrum – kanali 2/68 ekran 8/46 visoke tačnost 17/57 crtanje 28/49 stek 33/31

PC DOS tehnike

Device driver – 78/43
Ekran – Hercules 21/38 55/52 57/58 YU 76/40
Fajlovi – 56/48 57/68 63/44 81/72 99/48
DOS ispod haube – 24/18 50 77/50 80 94/58
IOCTL 74/51
Editor (pisanje) – 80/52 81/57 83/48 94/54
Environment – 96/70
Korisnički interfejs – 71/57
Makro naredbe – 44/76
Miš – 93/62
Mreže – 68/55 81/54
Nove naredbe – 26/36
RAM disk – Životić Zoran
Rezidentni programi – 32/62, 73/51 85/52 93/64, 94/90
Startovanje programa – 25/32
Upravljanje memorijom – 45/58 LIM 60/36 72/55 50/70 DPMI 96/56 97/62 XMS 98/56
Modem (komunikacije) – 57/84 61/73 95/74, 97/64
Tastatura – 57/73 69/98
Zaštita – 74/38 75/54

PC Windows tehnike

Batch interpreter – 86/78
Bit mape – 97/56
Formulen – 76/42
Modem (upravljanje) – 85/56
ObjectVision – 71/29
Object Win Library – 99/60
Osnove – 73/54 79/54 88/72
Prozori i poruka – 74/68 75/56
Scroll – 77/58

Programiranje igara

Akcione igre – 2/72
Logičke igre – 9/53 11/52 13/54 41/55 43/43
Loto – 20/62 21/55 23/48 25/54 26/34 27/48 28/47 29/44 31/59 33/27 34/38 40/58 44/29 46/62 53/90 62/54
Poker bez karata – 11/32
Veštačka inteligencije – Eliza 5/53 GUS 25/24, 32/28 36/48

Put u središte ROM-a

Put u središte ROMa – 4/69 raspodela memorije 5/41 video memorije 8/41 bejzik interpreter 7/72 editor 9/37 procedure 12/52 izračunavanje izraza 14/54 predstavljanje brojeva 18/56 struktura asemblera 25/56 29/56 simulator 34/30 disasembler 35/52 monitor 37/62

TEORIJA PROGRAMIRANJA

Algoritmi

Hanojske kule – 20/50
Holandska zastava – 12/20
Kalendar – 94/79
Kombinatorika – 24/24
Kriptografije – 12/32 78/38 79/44
Lavirint – 13/47
Metropolis – 92/41
Najkraći put – 11/37
Opšte – 8/49 33/59 34/56 49/115 35/74 36/68 81/48 83/38
Popunjavanje konture – 17/35
Rastavljanje reči na slogove – 18/52

Kompresije

Algoritmi – 80/40 95/52 LZW 58/62 60/43
Hafmenovi kodovi – 48/60 96/60
Slike – 98/40
Tekst – 12/28

Matematičke metode

Diferenciranje – 80/59
Elementarne funkcije – stepena 16/60 kvadratni koren 9/44 eksponencijalna 10/54 logaritamska 12/44 trigonometrijske 14/60 area 13/44 15/60 59/60 hiperboličke 11/55 17/82 60/62
Geometrijsko modeliranje – 75/44
Lagranžovi polinomi – 11/50
Metodi – gradijenta 24/54 tangente 25/48 sečice 26/54 intervala 27/54 asimptote 50/78 eliminacije 28/60 matrica 29/54 iteracija 31/52 subtabulacije 32/38 interpolacije 9/60 33/54 15/82 aproksimacije 34/24 ekonomizacije 45/50
Modeliranje gravitacionog polja – 61/81
Trapezno pravilo – 10/46
Računarski poentilizam – 57/74

Moderne programerske tehnike

Debagovanje programa – 91/70 97/58
Objektno programiranje – 36/28 73/36 92/84
Optimizacija – 60/58
Strukturirano programiranje – 4/35

Pretraživanje

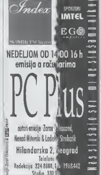
Algoritmi – 10/52 13/32 16/57 17/52 21/50 23/50 28/42 89/78 90/68 91/63 92/56
Baze teksta – 94/64
Binarno stablo – 26/38
Heurističko pretraživanje – 39/30 40/56, 41/55 43/43

Strukture podataka

Elementarne – 85/44
Liste – 36/38 37/30 39/28 40/33 41/48 60/48
Stabla – 42/44 43/48 44/74

Sortiranje

Algoritm – 18/56, 20/56, 30/60, 41/53, 44/90 86/71 88/90
Praksa – 67/32
YU sortiranje – 23/40



INFORMACIJE

AKTUELNOSTI

Svet

Razglednica iz Londona – 2/6 4/5 5/8 7/5 7/8 9/7 11/5 12/4 17/10 18/2 18/7 19/2 21/2 22/2 23/2 25/2 32/7
Razglednica iz Njujorka – 9/5 13/16, 14/5 16/10 17/8 20/2 22/4 21/10 24/2 27/12 52/6 55/4
Sajmovi – 13/12 26/9 28/14 32/7 36/8 39/12 44/10 45/7 49/7 55/5 65/59 66/36 86/8 72/20 77/8 92/10 83/19 99/31
Trendovi – 5/11 21/18 31/8 34/9 64/24 64/28 72/7 72/8 73/16 74/8 75/6 86/14 93/11 98/45 98/54
Zanimljivosti i informacije – 3/22, 3/5 10/44 21/8 23/17 54/64 56/4 96/62

Zemlja

Firme i događaji – 10/18, 52/10 53/8 66/30 67/8 96/8
Kodni rasporedi – 47/58 Unicode/Uniham 79/22 CP852 64/22 90/51 UYU 58/58, tastatura 48/76 73/68
Klubovi programera – 1/61 1/85 10/44 32/6 47/14
Sajmovi – 21/23 33/14 45/12 48/15, 62/6, 66/8 77/49 87/81 93/17
Školstvo – 1/66 2/40 6/53 14/16, 15/20 31/56, 64/29 65/64
Zakoni – 11/24 20/24 21/22 23/23

INTERVJUI

Dragan Antonijević – 31/57
Branislav Bakić – 12/14
Dejvid Banel – 42/8
Milan Božić – 50/75 88/32
Jovan Čiplov – 66/38
Rajko Dujmić – 52/14
Jozo Dujmović – 5/13
Stiven Džobs – 7/13
Igor Fischer – 5/8
Majkl Frutig – 60/9
Bill Gates – 22/12
Sonja Golomeic – 57/4
Endi Hercfeld – 25/12
Toru Ivanlai – 21/14
Garry Kildal – 23/14
Skot Kim – 28/14
Dragan Kopunović – 41/14
Vlada Kostić – 8/24
Pol Krneta – 95/38 97/7
Fred Langa – 90/20
Mirko Ležaj – 51/48
Žika Manzalović – 56/22
Vlastimir Matejić – 93/20
Radomir Mihajlović – 9/15
Draško Milićević – 48/6
Lim Hue Meng – 64/27
Milan Mijić – 90/52
Vojko Mlakar – 45/10
Čeda Mraz – 43/10
Janko Mršić Flogel – 36/11
Rej Ouzi – 28/18
Branislav Pekić – 64/6
Milosav Slavko Pešić – 70/4
Džef Raskin – 24/14
Vejn Ratlif – 27/14
Džonatan Saks – 26/22
Slobodan Selenić – 65/4
Dušan Slavić – 8/12
Janez Škrubej – 34/6
Kruno Toverlić – 47/12
Nebojša Vasiljević – 31/58
Vidoje Vasiljević – 44/8
Chris Wells – 73/38

ISTORIJA RAČUNARSTVA

Alan Tjuring – 11/57
Bil Gates – 84/36 88/0
Bill Gates o istoriji bejzika – 57/14
Četrdeset godina prvog računara – 15/8
Deset ludih godina – 4/10
Deseti rođendan kompjuterskih virusa – 97/69
IBM – 12/17
Istorija DOS-a 61/20
Istorija Unixa – 71/43
Pedeset godina HP – 56/12
Prvi programer u istoriji – 1/84
Rađanje programskih jezika – 5/20
S. Crey – 68/12
Šta smo kupali kad smo bili mali – 86/52